MAXTRAC/RADIUS

In typical Motorola fashion, the Maxtrac and Radius line of mobiles are really the same radio! The only real difference is the actual firmware of the radio. All of the info provided here should work with varying degrees of success depending on the features of the original radio. The Service Manual is crammed full of stuff and would be well worth ordering if you plan to do any experimenting with the radio. The part number of the Service Manual is 6880102W84 for the Maxtrac and it contains info on all models of the radio (Low, VHF, UHF, and 800 MHz).

Do you need a pinout for the 16 pin accessory connector, the 5 pin accessory connector, or the mic connector, or how about the programming cable?

If you want to inject audio into your Maxtrac/Radius while tuning it up with the RSS, you will want to have a look at this schematic.

For a easy overview of all the programmable pin configurations on the accessory connector, check out this Excel Spreadsheet or this image.

If you want an easy way to connect to the Accessory Connector on these radios, you can order the HLN9242A or HLN9457A Accessory Connector kits. These include an accessory connector housing, some pins, and some wires with pins already attached. One thing to note, the HLN9242A kit also includes a pin extractor for removing previously installed pins.

If you would rather build your own accessory cables, the parts are available from DigiKey. You will want the following:

- Connector, with locking tab 16 pin, 104422-1-ND (AMP 104422-1)
- Contact, A3007-ND(AMP 1-87309-3)
- Housing, 5 pin, 455-1186-ND (AMP VHR-5N)
- Connector terminal crimp, 455-1319-1-ND (AMP SVH-41T-P1.1)

The first two parts are for a 16 pin accessory connector. The second two are for a 5 pin accessory connector.

Do you want to program frequencies outside of the normal bandsplit? For a Maxtrac look at Maxtrac, for a Radius look at Radius.

Having RSS errors? Look at the Radius Series RSS Error Codes

Some people's voices just suck when broadcast over a Maxtrac/Radius due to the the cutoff frequencies used in the TX Audio stage. Here is modification to make the audio more pleasing to the ear.

If you have a TDN8310A DTMF microphone from a Maxtrac and are looking for a schematic for it... look no further. The proper source for the schematic is in the 6880309C84 manual. However, this microphone is almost identicle to the VLN1017A DTMF microphone for the MCX1000. You can look at the Tone/Logic Board and the Microphone/Beeper Board schematics. Note that the part designations in these schematics are slightly different for the TDN8310A, but it is better than nothing. This board layout will make locating the proper jumpers a bit easier.

If you get the above mentioned 6880309C84 manual, you will find that it covers the following microphones:

- TDN8305A/B
- TDN8306A/B
- TDN8307A/BSP
- TDN8309A/B

- TDN8310A/B

Another microphone manual that may be of interest is the 6881085E15. It covers the Auto-Dial Touch Code Encoder Palm Microphones with Backlit Keypad. These are microphones:

- TMN6169A - DTMF Maxtrac Conventional
- TMN6170A - DTMF Maxtrac Trunked
- TMN6171A - DTMF MaraTrac A2, A3
- TMN6172A - DTMF MaraTrac A7
- TMN6173A - DTMF Spectra Conventional
- TMN6174A - DTMF Spectra Trunked
- TMN6175A - DTMF Mostar Conventional
- TMN6176A - DTMF Mostar Trunked

One interesting feature of these microphones is they have an option jumper that allows the +12 volts of a Systems 9000 microphone connector to power the keypad backlight (you also need a Systems 9000 microphone cord with all 6 wires). These microphones have 9 auto-dial memories and one last number dialed memory (the manual has the programming info).

If you want to build a nice little programmer for the Maxtrac/Radius radios, one that does not require a RIB, check out this link. Just connect the BUS+ and GND wires from the circuit to the appropriate pins on the microphone connector as shown in the programming cable schematic. If you check the RIB page you will find some additional circuits you can try.

Identifying Model Variations

One thing difficult about this line of radios, it determining what radio you have in your hands without being able to read it with the appropriate RSS. This is because there is only so much information you can gain from the model number (namely band and power).

First a little bit about the model number. If you look at a typical model number, D35LRA5GB6BK, the quickest way to tell what sort of radio you have is to look at the second and third characters. They determine output power level and frequency range, respectively:

Second Character	Description
0	2 Watts
2	10 Watts (900MHz)
3	25 Watt (VHF/UHF) 15 Watt (800MHz)
4	40-50 Watt (VHF) 40 Watt (UHF) 35 Watt (800/900MHz)
5	60 Watt (Low Band)

Third Character	Description
1	Low Band (29.7-36.0MHz, 36.0-42.0MHz, 42.0-50.0MHz)
3	VHF (136-162MHz or 146-174MHz)
4	UHF (403-430MHz or 449-470MHz)
5	800MHz
7	900MHz

Now, in order to narrow down the actual frequency split without reading the radio. You will have to disassemble the radio, remove the top shield on the RF board and look for the part number stamped or silk screened on the PC board. From the following chart, you should be able to determine the bandsplit of the radio:

Part Number	Bandsplit

HLB4099A	29.7-36.0MHz
HLB4100A	36.0-42.0MHz
HLB4101A	42.0-50.0MHz
HLD4321B	136-162MHz
HLD4322B	146-174MHz
HLE4424A(B)	403-430MHz
HLE4425A(B)	449-470MHz
HLE9310A(B)	449-470MHz
HLF4095B	RX 851-870MHz TX 806-825MHz
HLF9122A	RX 851-870MHz TX 806-825MHz TX 851-870MHz

The only thing left is to figure out which logic board you have. There are four different ones used in the Maxtrac series. The part number can be determined while you already have the radio apart, just locate the part number stamped on the logic board (bottom of the radio).

Part Number	**Description**
HLN9313A	Full Options and Signalling Capable, 16 Pin Accessory Connector, Expanded Board
HLN9123A	Basic Limited Board, No Signalling, 5 Pin Accessory Connector, Limited Channels, Masked Board
HLN5173B	Signalling Capable, 5 Pin Accessory Connector, No External EEPROM (Limited Channels), Expanded Board
HLN5172A	Same as HLN9313A, Except 5 Pin Accessory Connector

The logic boards can be found in just about any bandsplit radio. The logic board is not frequency dependent and it operation is determined by what firmware is installed in it (expanded boards), as well as what it was initialized with in the RSS.

This should give you a start on determining what radio you have in your hands. The only real way to determine the what all the options the radio is currently capable of is to read it. If you are going to be hacking on these radios, you would want to try and get one with either a 5172 or 9313 logic board in it, unless you are just building a repeater link or something.

Maxtrac/Radius Power Problems

Sometimes you may come across one of these radios which will not power up when you apply 12VDC to the power connector on the back of the radio. Well in this case, one of the first things you should check is that the radio has not been setup for Ignition Control. If it is, then the radio will not power up until you apply 12VDC to the appropriate Ignition Control pin on the Accessory Connector.

The quickest way to check to see if this is the case is to use an ohm meter and measure for continuity between the positive connection on the main power connector and Pin 5 of a 5 pin accessory connector, or Pin 10 of the 16 pin accessory connector. If you have continuity, then Ignition Control is not your problem... get out the service manual. If, on the other hand, you do not measure continuity, then read on!

You now have a choice, you can either continue to use the radio in this fashion, by getting the appropriate accessory connector for your radio and applying 12VDC to the Ignition Control pin to turn the radio on. Or, you can restore the radio to normal operation by replacing the missing component in the radio.

To restore the radio to normal operation, you will need to remove the head and bottom cover from the radio. This should expose the logic board. Now, locate the component that you are interested in, it is F801 for the 5 pin type logic board, or JU801 on the 16 pin logic boards. These devices are a 1A and 2A fuse in a 1/4W resistor type package. If the one in your radio is missing or burnt out, your radio will not power up without power applied to the accessory connector. It is quite common for the fuse to be blown when the pin on the Accessory Connector is shorted to ground. That is why when you replace or install this component, you must use the appropriate fuse, and not just a piece of wire, unless you want to replace your logic board in the future.

Once you replace this component, the radio should power up as it normally would.

Linking Maxtracs for Repeater/Crossband Use

Need to connect a couple of Maxtrac/Radius radios together? Here is a schematic of the R.I.C.K. to get you started. Order the service manual part number 6880901Z79 from Motorola.

An alternative method of linking 2 Maxtrac/Radius type radios together can be found here. **DO NOT USE THIS WITH 900MHz RADIOS, the pinout of the microphone connector is different!**

Trunking 800MHz Maxtracs

These are radios that usually have model numbers starting with D35 or D45.

You can convert a trunking Maxtrac or Radius back to conventional 800MHz operation. You will need to replace the trunking firmware in the radio with conventional firmware. You will then need to blank the logic board with Lab RSS, and re-initialize it as a conventional radio. Note that before changing the firmware and blanking the board, you should write down all the alignment settings in the radio.

Conventional Maxtrac's make nice receivers for receiving your local trunking system's data channel... for use in Trunker of course. If you don't know what Trunker is, you better check out Lindsay's Site. You will find that if you connect your data slicer to RX Audio (Pin 11) and Ground (Pin 7) on the Accessory Connector, you will have a nice little setup. One thing you will want to check though, is that JU551, a 3 pin jumper header on the logic board, is set to Position A. This will give you flat, unmuted audio on Pin 11 of the Accessory Connector. This is assuming of course that you are using a radio with a 16 pin Accessory Connector, otherwise, you are on your own.

If you're playing with Trunking Maxtracs and TrunkTrackers, you may be interested in this conversion spreadsheet. It is a Microsoft Excel document. It converts Motorola Talkgroup ID's to Maxtrac ID's and TrunkTracker ID's.

If you are working on tuning up a Trunked Maxtrac, here is the information on putting the radio into test mode.

900MHz Amateur Radio Conversion

The Maxtrac/Radius will work just fine in the Ham band. But, you will need to make some modifications to the RF section as well as some of the programming in order to get it to work properly.

Before you go any further, you must be trying to convert a 900MHz radio, these would be models that have model numbers starting with either D27 or D37. If you have a D35 or D45 radio, then you have an 800MHz radio, and you will not be able to convert these radios to the 900MHz band.

You can convert the trunking D27 "B2" radio, and make it into a 2 channel conventional radio.

If you want the straight forward instructions on converting the radio, then follow this link. The modifications presented there should provide sufficient results for most people.

If you are looking for an explanation on how the Pin Shift signal works and which VCO's do what, then you will want to look at the Pin Shift explained page.

If you are a die hard and want full coverage on transmit and receive, there are going to be some more modifications involved. You will want to investigate these modifications which came from Michael Roche, AA2LS.

You can also get a little help with some of the tune-up instructions if you check here.

50-54MHz (6 Meters) Amateur Radio Conversion

You've got a Maxtrac/Radius that you want to tune up on 6m... check here.

9600 Baud Packet Conversion

The Maxtrac/Radius Series are 9600 baud packet capable. They do need a minor modification though. For RX, its no big deal, just take RX audio from PIN 11 of the Accessory Connector and swing JU551 to the (A) position (Flat RX Unmuted). For TX audio, you will have to run a jumper to the junction of R223, C228, R222 on the RF board to inject the data (if applicable you may need to set the deviation pot in the radio to maximum). You might want to put a pot on your TX line somewhat like the setup Motorola uses (voltage divider) to adjust the deviation. It would be wise to do some measurements to make sure you have the deviation correct. You might also want run the RX and TX pins to the unused pins on the front mic connector after the mods are done.

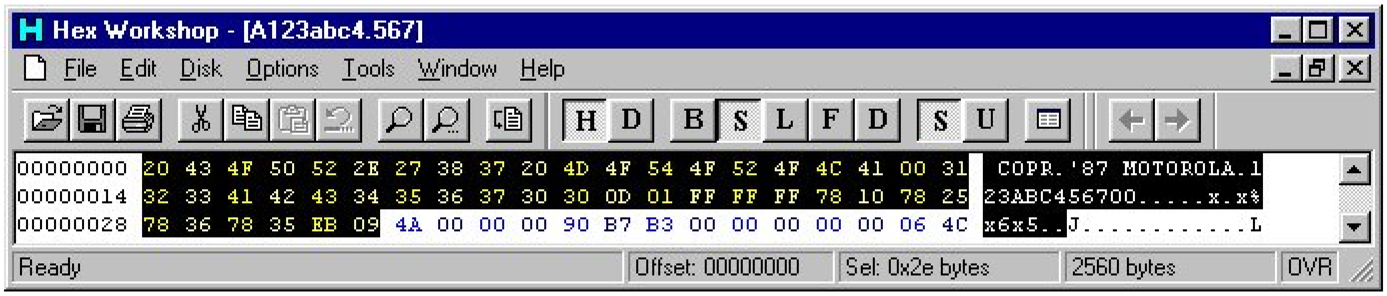
Information Common to Both Maxtrac and Radius

Since these radios are so much alike, lets start with some things you can do that apply to both series of radios.

Codeplug Checksum Location

If you happen to be messing around in the codeplug file, be aware that the first 0x2D bytes need to have the proper checksum (stored at offset 0x2E).

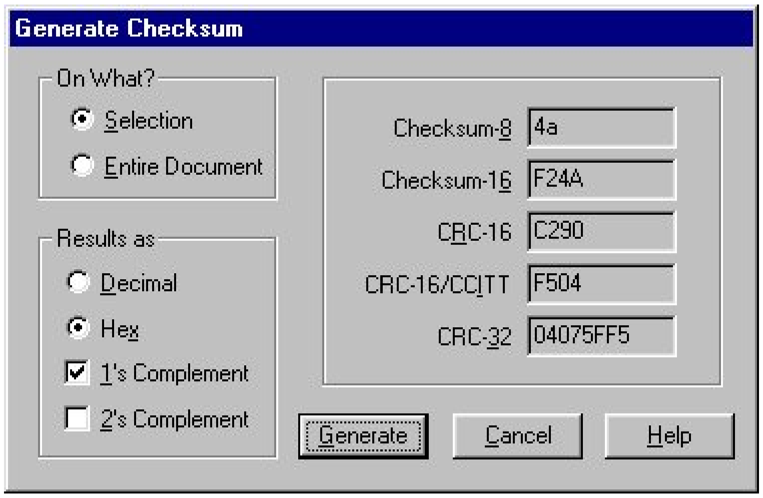
For example, if you wanted to calculate the checksum of the following header in a codeplug file:


Hex Workshop - [A123abc4.567]
File Edit Disk Options Tools Window Help
H D B S L F D S U

```
00000000 20 43 4F 50 52 2E 27 38 37 20 4D 4F 54 4F 52 4F 4C 41 00 31  COPR.'87 MOTOROLA.1
00000014 32 33 41 42 43 34 35 36 37 30 30 0D 01 FF FF FF 78 10 78 25 23ABC456700.....x.x%
00000028 78 36 78 35 EB 09 4A 00 00 00 90 B7 B3 00 00 00 00 00 06 4C x6x5..J...........L
```

```
Ready | Offset: 00000000 | Sel: 0x2e bytes | 2560 bytes | OVR
```

You would select the bytes and do a 1's Complement Hex Checksum calculation.


Generate Checksum

On What?
- [✓ Selection]
- Entire Document

Results as
- Decimal
- [✓ Hex]
- [✓ 1's Complement]
- 2's Complement

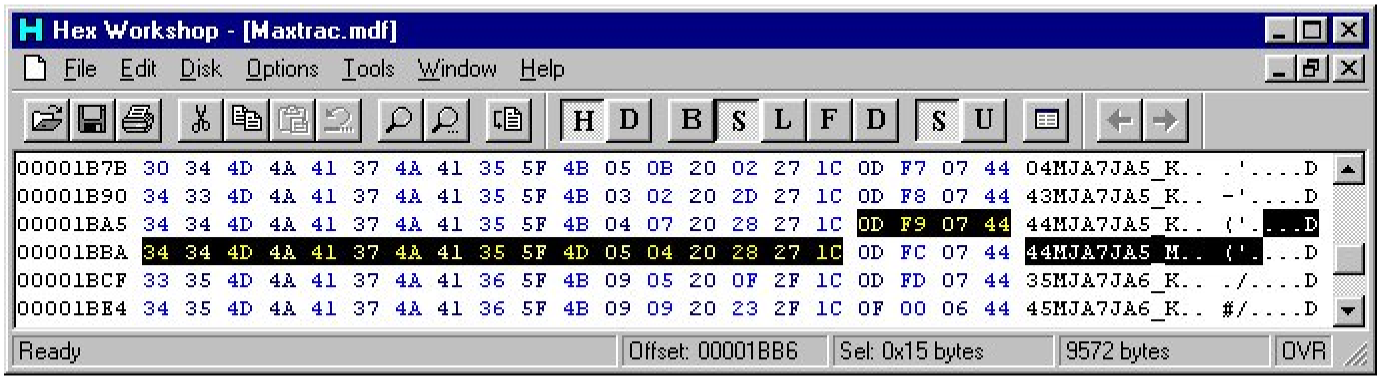
Checksum-8: 4a
Checksum-16: F24A
CRC-16: C290
CRC-16/CCITT: F504
CRC-32: 04075FF5

The Checksum-8 value is the one you want to enter into offset 0x2E. In this case you can see that they already match.

We've been told that the same applies if you are changing the serial number of the radio using the EEPROM access feature of Lab RSS. The nice feature of using the Lab RSS is that it has the option to correct the checksum for you.

Location of Model Index in Codeplug

First, lets take a look at an example from a .mdf file:

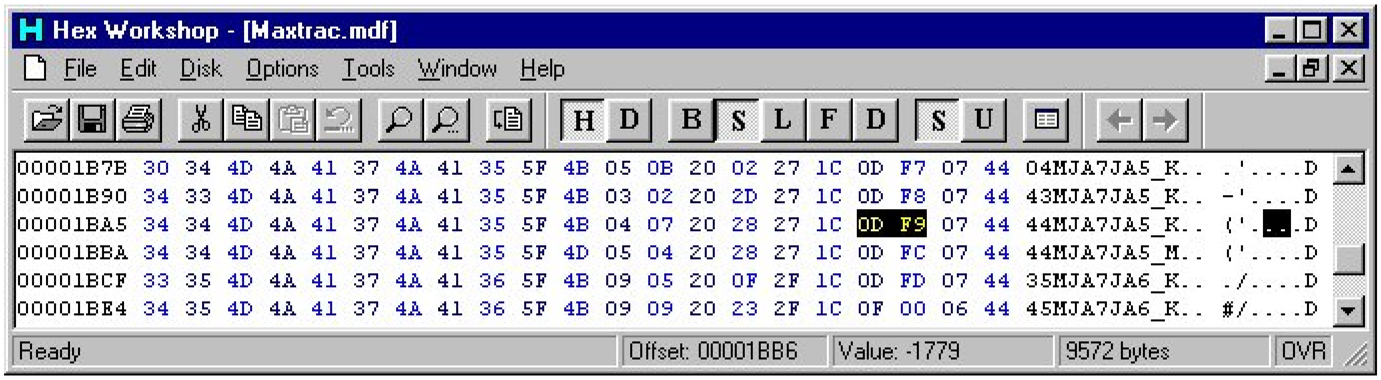

Hex Workshop - [Maxtrac.mdf]

```
00001B7B 30 34 4D 4A 41 37 4A 41 35 5F 4B 05 0B 20 02 27 1C 0D F7 07 44 04MJA7JA5_K.. .'....D
00001B90 34 33 4D 4A 41 37 4A 41 35 5F 4B 03 02 20 2D 27 1C 0D F8 07 44 43MJA7JA5_K.. -'....D
00001BA5 34 34 4D 4A 41 37 4A 41 35 5F 4B 04 07 20 28 27 1C 0D F9 07 44 44MJA7JA5_K.. ('....D
00001BBA 34 34 4D 4A 41 37 4A 41 35 5F 4D 05 04 20 28 27 1C 0D FC 07 44 44MJA7JA5_M.. ('....D
00001BCF 33 35 4D 4A 41 37 4A 41 36 5F 4B 09 05 20 0F 2F 1C 0D FD 07 44 35MJA7JA6_K.. ./....D
00001BE4 34 35 4D 4A 41 37 4A 41 36 5F 4B 09 09 20 23 2F 1C 0F 00 06 44 45MJA7JA6_K.. #/....D
```

Ready | Offset: 00001BB6 | Sel: 0x15 bytes | 9572 bytes | OVR

The highlighted entry contains all the info for a specific model of radio.

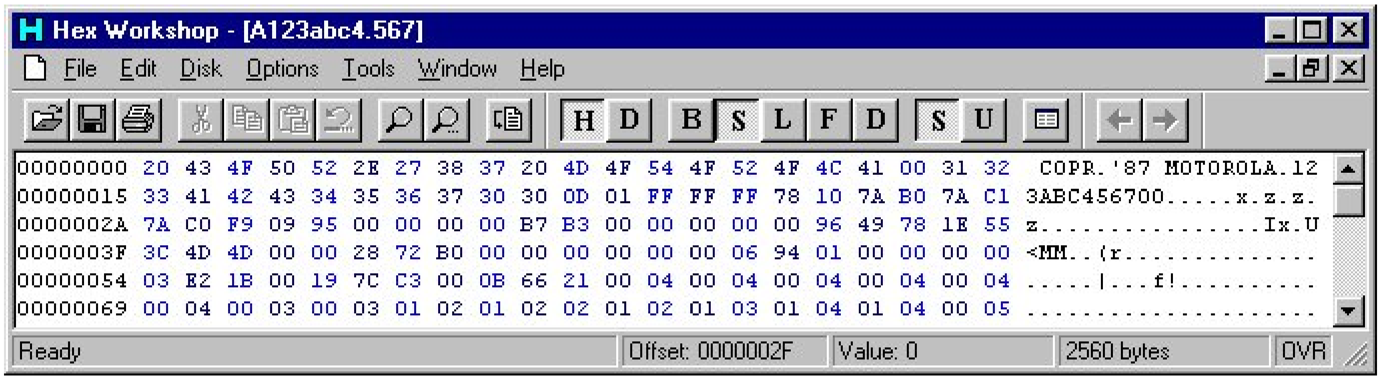
For the particular model selected, the index that refers to this radio is this:


Hex Workshop - [Maxtrac.mdf]

```
00001B7B 30 34 4D 4A 41 37 4A 41 35 5F 4B 05 0B 20 02 27 1C 0D F7 07 44 04MJA7JA5_K.. .'....D
00001B90 34 33 4D 4A 41 37 4A 41 35 5F 4B 03 02 20 2D 27 1C 0D F8 07 44 43MJA7JA5_K.. -'....D
00001BA5 34 34 4D 4A 41 37 4A 41 35 5F 4B 04 07 20 28 27 1C 0D F9 07 44 44MJA7JA5_K.. ('....D
00001BBA 34 34 4D 4A 41 37 4A 41 35 5F 4D 05 04 20 28 27 1C 0D FC 07 44 44MJA7JA5_M.. ('....D
00001BCF 33 35 4D 4A 41 37 4A 41 36 5F 4B 09 05 20 0F 2F 1C 0D FD 07 44 35MJA7JA6_K.. ./....D
00001BE4 34 35 4D 4A 41 37 4A 41 36 5F 4B 09 09 20 23 2F 1C 0F 00 06 44 45MJA7JA6_K.. #/....D
```

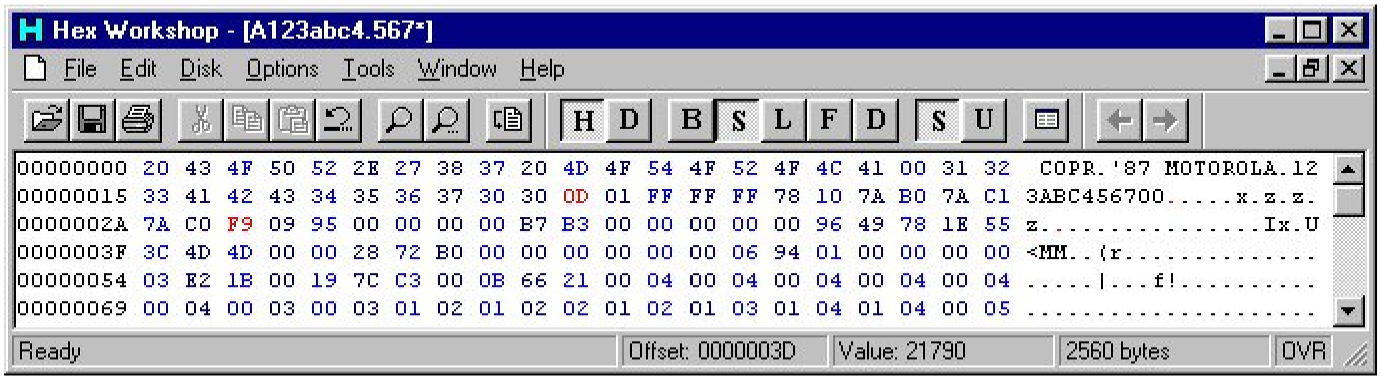
Ready | Offset: 00001BB6 | Value: -1779 | 9572 bytes | OVR

Now, if you open the saved file (codeplug) for your radio, you'll see something like this at the beginning of the file:



```
Hex Workshop - [A123abc4.567]
00000000 20 43 4F 50 52 2E 27 38 37 20 4D 4F 54 4F 52 4F 4C 41 00 31 32  COPR.'87 MOTOROLA.12
00000015 33 41 42 43 34 35 36 37 30 30 0D 01 FF FF FF 78 10 7A B0 7A C1 3ABC456700.....x.z.z.
0000002A 7A C0 F9 09 95 00 00 00 00 B7 B3 00 00 00 00 00 96 49 78 1E 55 z...............Ix.U
0000003F 3C 4D 4D 00 00 28 72 B0 00 00 00 00 00 06 94 01 00 00 00 00 <MM..(r..............
00000054 03 E2 1B 00 19 7C C3 00 0B 66 21 00 04 00 04 00 04 00 04 00 04 .....|...f!..........
00000069 00 04 00 03 00 03 01 02 01 02 02 01 02 01 03 01 04 01 04 00 05 .....................
Ready                          Offset: 0000002F    Value: 0         2560 bytes   OVR
```

Now, the tricky part of this is, the index isn't stored in one location. The red bytes are the index. Notice they match the index bytes you found in the .mdf file.



```
Hex Workshop - [A123abc4.567*]
00000000 20 43 4F 50 52 2E 27 38 37 20 4D 4F 54 4F 52 4F 4C 41 00 31 32  COPR.'87 MOTOROLA.12
00000015 33 41 42 43 34 35 36 37 30 30 0D 01 FF FF FF 78 10 7A B0 7A C1 3ABC456700.....x.z.z.
0000002A 7A C0 F9 09 95 00 00 00 00 B7 B3 00 00 00 00 00 96 49 78 1E 55 z...............Ix.U
0000003F 3C 4D 4D 00 00 28 72 B0 00 00 00 00 00 06 94 01 00 00 00 00 <MM..(r..............
00000054 03 E2 1B 00 19 7C C3 00 0B 66 21 00 04 00 04 00 04 00 04 00 04 .....|...f!..........
00000069 00 04 00 03 00 03 01 02 01 02 02 01 02 01 03 01 04 01 04 00 05 .....................
Ready                          Offset: 0000003D    Value: 21790     2560 bytes   OVR
```

The significance of knowing the index number for your radio comes to when you start modifying the .mdf file for specific features in your model of radio. If you were to search for your model number in the .mdf file, you will have more than one hit, but, they each have unique index numbers...

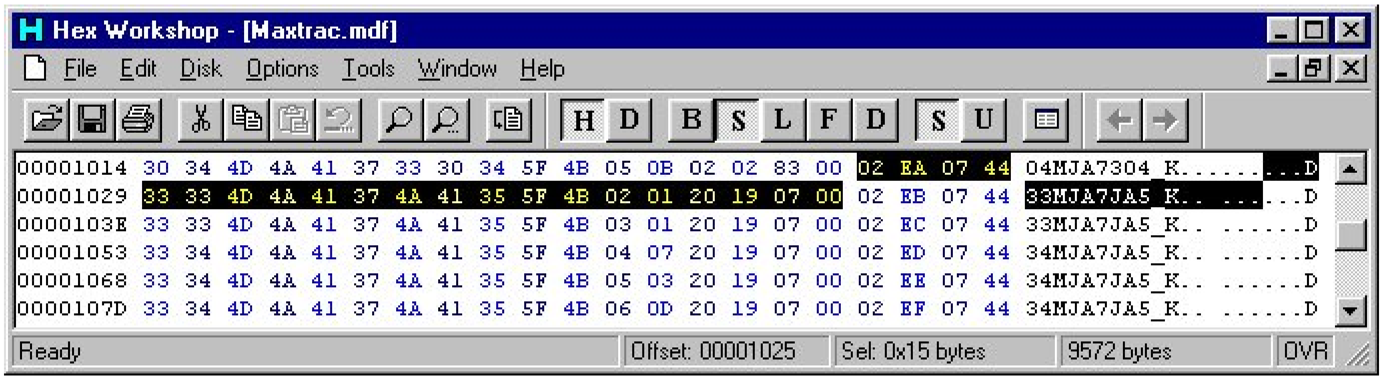
Add Channels by Editing the .MDF File

This information is common to both the Maxtrac and Radius lines. Depending on which logic board and firmware is in your radio, this may or may not work. The example screenshots shown are for the Maxtrac version R06.01.00c RSS.

Open the MAXTRAC.MDF or RADMBL.MDF in your Hex editor, Hex Workshop is an excellent program for this. The Maxtrac and Radius RSS don't do any checksum checks of the file so you don't have to worry about correcting the checksum later.

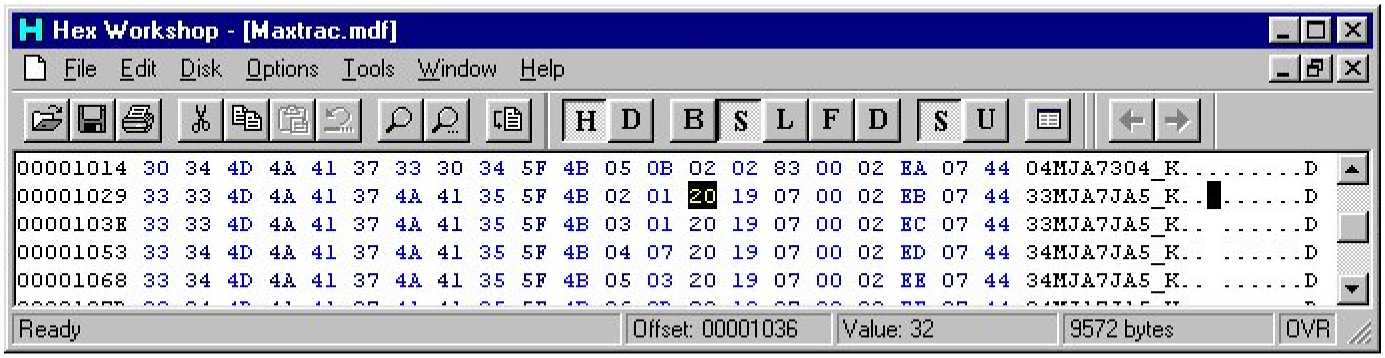
Search for your model number in the file, you probably will come accross more than one entry so you can either change them all or use the information above to locate the proper index to change a specific model.

Once you find your model entry you should see something like this:



```
Hex Workshop - [Maxtrac.mdf]
00001014 30 34 4D 4A 41 37 33 30 34 5F 4B 05 0B 02 02 83 00 02 EA 07 44 04MJA7304_K.......D
00001029 33 33 4D 4A 41 37 4A 41 35 5F 4B 02 01 20 19 07 00 02 EB 07 44 33MJA7JA5_K.. ....D
0000103E 33 33 4D 4A 41 37 4A 41 35 5F 4B 03 01 20 19 07 00 02 EC 07 44 33MJA7JA5_K.. ....D
00001053 33 34 4D 4A 41 37 4A 41 35 5F 4B 04 07 20 19 07 00 02 ED 07 44 34MJA7JA5_K.. ....D
00001068 33 34 4D 4A 41 37 4A 41 35 5F 4B 05 03 20 19 07 00 02 EE 07 44 34MJA7JA5_K.. ....D
0000107D 33 34 4D 4A 41 37 4A 41 35 5F 4B 06 0D 20 19 07 00 02 EF 07 44 34MJA7JA5_K.. ....D
Ready                          Offset: 00001025    Sel: 0x15 bytes  9572 bytes   OVR
```

If you look at this model entry, the third byte after the model number (offset 0x1036) is a 0x20.


Hex Workshop - [Maxtrac.mdf]
File Edit Disk Options Tools Window Help

```
00001014 30 34 4D 4A 41 37 33 30 34 5F 4B 05 0B 02 02 83 00 02 EA 07 44 04MJA7304_K.........D
00001029 33 33 4D 4A 41 37 4A 41 35 5F 4B 02 01 20 19 07 00 02 EB 07 44 33MJA7JA5_K.. ......D
0000103E 33 33 4D 4A 41 37 4A 41 35 5F 4B 03 01 20 19 07 00 02 EC 07 44 33MJA7JA5_K.. ......D
00001053 33 34 4D 4A 41 37 4A 41 35 5F 4B 04 07 20 19 07 00 02 ED 07 44 34MJA7JA5_K.. ......D
00001068 33 34 4D 4A 41 37 4A 41 35 5F 4B 05 03 20 19 07 00 02 EE 07 44 34MJA7JA5_K.. ......D
```

```
Ready                    Offset: 00001036    Value: 32     9572 bytes    OVR
```

This is the hex representation for the number of channels in the radio, in this case 32 ch. Depending on your radio you might want to try changing this to 0x10, 0x20, or 0x28, or higher to give you 16, 32, and 40 channels respectively.

Also note the 0x02 at offset 0x1034, this is the bandsplit identifier for this radio and corresponds to the table entry at the top of the .mdf file.

The 0x01 at offset 0x1035 refers to the power level of the radio.

To find out if you picked the correct model number to edit, save the .mdf file and load your radio's codeplug into the RSS. Try adding channels with the Mode Utility and see what happens.

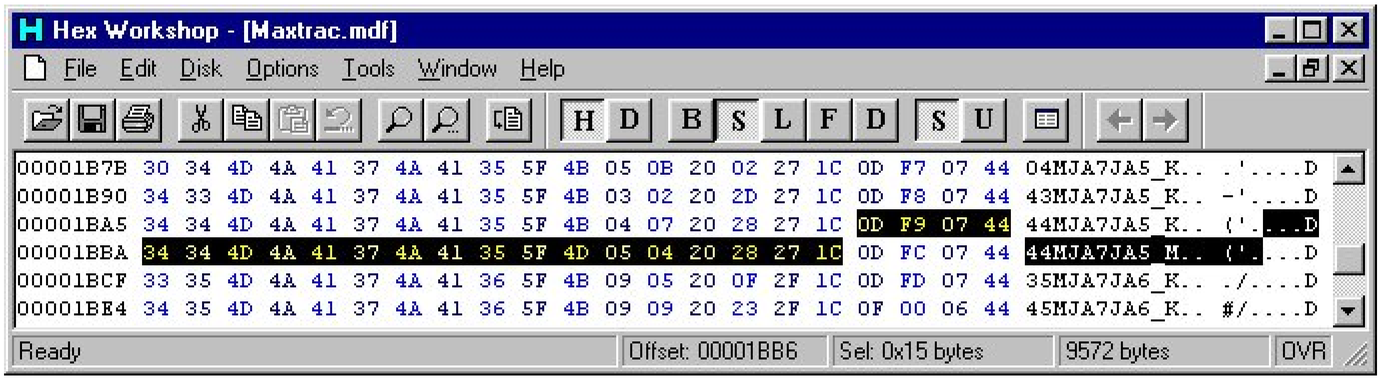
One thing you have to be careful of is that the RSS will let you put as many channels into it that you want, what the radio will accept is another thing. It appears that once you exceed the maximum number of channels for a radio and try to program them in, the firmware will "wrap" the extra data around to the beginning and start programming from channel 01. You will be able to tell if you have exceeded the maximum number of channels by trying to program in one channel at a time until it "wraps", ie. if the maximum number of channels your radio will take is 16 and you try to program 17 channels into it, once programming has finished you will find your radio will only show 1 channel. Another thing that might happen is that if you exceed the number of channels programmed into the radio, the display will show funny things as you scroll through the channels, it depends on the firmware in the radio.

.MDF File Structure

OK, we have finally figured out most of the information contained in a record in the .mdf file for the Maxtrac/Radius line.

By knowing this, we are able to do things like enable scan, signalling, etc. However, even if you enable the options by editing the .mdf file, it does not mean that they will always work, the firmware in the radio has to support the options as well.

Now, on to the structure itself. The highlighted area below indicates an individual record for a particular model Maxtrac. The Radius is essentially the same, just with different model numbers.


Hex Workshop - [Maxtrac.mdf]
File Edit Disk Options Tools Window Help
H D B S L F D S U

```
00001B7B 30 34 4D 4A 41 37 4A 41 35 5F 4B 05 0B 20 02 27 1C 0D F7 07 44 04MJA7JA5_K.. .'....D
00001B90 34 33 4D 4A 41 37 4A 41 35 5F 4B 03 02 20 2D 27 1C 0D F8 07 44 43MJA7JA5_K.. -'....D
00001BA5 34 34 4D 4A 41 37 4A 41 35 5F 4B 04 07 20 28 27 1C 0D F9 07 44 44MJA7JA5_K.. ('....D
00001BBA 34 34 4D 4A 41 37 4A 41 35 5F 4D 05 04 20 28 27 1C 0D FC 07 44 44MJA7JA5_M.. ('....D
00001BCF 33 35 4D 4A 41 37 4A 41 36 5F 4B 09 05 20 0F 2F 1C 0D FD 07 44 35MJA7JA6_K.. ./....D
00001BE4 34 35 4D 4A 41 37 4A 41 36 5F 4B 09 09 20 23 2F 1C 0F 00 06 44 45MJA7JA6_K.. #/....D
Ready | Offset: 00001BB6 | Sel: 0x15 bytes | 9572 bytes | OVR
```

As you can see, this example covers a model located between offsets 0x01BB6 and 0x01BCA inclusive. The breakdown of the record is as follows:

Offset (example)	Description
0x1BB6-0x1BB7	Model Index
0x1BB8	Displayed Model Name
0x1BB9-0x1BC4	Model Number
0x1BC5	Bandsplit Index
0x1BC6	Radio Power Level
0x1BC7	Number of Channels
0x1BC8	Displayed Power
0x1BC9-0x1BCA	Radio Options

<u>Field descriptions:</u>

Model Index

As can be seen elsewhere on this page, the Model Index is what actually tells the RSS what radio you are working with. It figures out the index from the codeplug in the radio or on file and locates the appropriate record in the .mdf file based on that information.

Once the RSS knows the Model Index, it can read all the specific information on what that radio is capable of by reading the rest of the model information in the .mdf file record.

Displayed Model Name

This byte determines what model name (ie. Maxtrac 300, Radius M214/216) is displayed in the RSS when you read the radio codeplug.

All the possible model names are stored at the beginning of the .mdf file. If you change this byte, the model name displayed will change. For example, this radio has a byte of 0x07. If you read this radio, it will have a name of Maxtrac 300. If you change the byte to 0x06, the name will change to Maxtrac 100.

Model Number

These bytes are the hex representation of the actual model number of the radio in question. That's about it. There are a number of radios in the Maxtrac/Radius series that have the exact same model number, but have different features or options. That is why there is no model breakdown chart for them. The only way to actually figure out the options in a particular radio is to read it.

Bandsplit Index

This byte determines which bandsplit information record is used for this radio. The bandsplit table is located just under the model names at the beginning of the .mdf file. More information on the structure of that table is available elsewhere on this page.

Radio Power Level

Not exactly sure on how this one works. It depends on what the bandsplit of the radio is and the power level (high or low).

Number of Channels

This byte is a hex representation of the number of channels in the radio. It is explained further elsewhere on this page.

Displayed Power

This byte is a hex representation of the power level of the radio. It is used to display the power level on the Radio Wide screen. By changing this byte, you will change the power field on the Radio Wide screen. It has no other effect.

Radio Options

These two bytes are some of the most important. They determine what options are enabled in the radio and are capable of being modified by the RSS. Now, just because you enable the options, it doesn't mean that your radio will support them, that depends on the firmware in the radio, as well as the codeplug in the radio. It is generally possible to upgrade the firmware in the radio, and usually will involve blanking, re-initializing, and re-tuning the radio in the process, but it is possible.

It also appears that some options need to be turned on in the codeplug of the radio, regardless of whether those bits are enabled in the .mdf file. The bytes in the radio that seem to affect these options are at offset 0x1FE and 0x1FF in a codeplug archive.

At any rate, what we have determined from the .mdf file is as follows...

Options are determined by whether a specific bit in the option bytes are turned on or off. Each of the two option bytes can be broken down into an upper nibble, and a lower nibble. When each nibble is converted to binary, you get all the possible option bits that can be changed.

In this example, our option bytes are the fairly common 0x27, 0x1C. This translates as follows:

```
NAME     UPPER1      LOWER1      UPPER2      LOWER2
HEX         2           7           1           C
BIN      0 0 1 0     0 1 1 1     0 0 0 1     1 1 0 0
BIT#     4 3 2 1     4 3 2 1     4 3 2 1     4 3 2 1
```

If we look at the UPPER1 nibble, bits 1 through 4 correspond to the following options when set (1):

```
BIT#     DESCRIPTION
1        Unknown, causes a ERROR #56 if set
2        Enables Emergency Alarm and Signalling (MDC1200/STAR/QCII/DTMF)
3        Unknown
4        Unknown
```

If we look at the LOWER1 nibble, bits 1 through 4 correspond to the following options when set (1):

```
BIT#     DESCRIPTION
1        TPL/DPL/INV DPL/CSQ Squelch, Squelch field under Radio Wide set to CODED
2        TPL/DPL/INV DPL/CSQ Squelch, Squelch field under Radio Wide set to CARRIER
```

```
          3        Enables Scan
          4        Unknown
```

Note, if you do not set either bit 1 or 2 of LOWER1, then the only squelch option you are allowed in a Maxtrac is INV DPL, and in a Radius, CSQ. Normally both bits 1 and 2 are always set.

If we look at the UPPER2 nibble, bits 1 through 4 correspond to the following options when set (1):

```
          BIT#      DESCRIPTION
          1         Enables Accessory Connector (if supported by radio)
          2         Unknown
          3         Unknown
          4         Unknown
```

If we look at the LOWER2 nibble, bits 1 through 4 correspond to the following options when set (1):

```
          BIT#      DESCRIPTION
          1         Unknown
          2         Unknown
          3         Affects TOT, base frequency, and signalling options, must be set
          4         Enable Call List (used in conjunction with UPPER1, BIT#2
```

To determine what to put in the .mdf file in the option bit locations, figure out what bits you want set, convert the binary representation of the nibble to hex, and stuff it in the proper location in the .mdf. In general, if you want to enable scan and signalling (if supported by your radio), use the option bytes in this example (0x27, 0x1C).

Codeplug Hacking

Well, if you are brave enough to mess around in the codeplug using the EEPROM Access feature of the Maxtrac Lab RSS, there are some things you should know.

With version R07.02.00a Maxtrac RSS and newer, the codeplug errors you may get when reading the radio have been expanded from the usual Error #58 into a bunch of different codes. Depending on what the code is, determines the block of data in the codeplug which is corrupted. The blocks we have identified so far are as follows:

- Address 0x000 to 0x02D, checksum at 0x02E, Error #58
- Address 0x02F to 0x039, checksum at 0x03A, not checked
- Address 0x03B to 0x07E, checksum at 0x07F, Error #59
- Address 0x080 to 0x08B, checksum at 0x08C, Error #60
- Address 0x08D to 0x093, checksum at 0x094, not checked
- Address 0x095 to 0x09B, checksum at 0x09C, not checked
- Address 0x09D to 0x0A3, checksum at 0x0A4, not checked
- Address 0x0A5 to 0x0AB, checksum at 0x0AC, not checked
- Address 0x0AD to 0x0B3, checksum at 0x0B4, not checked
- Address 0x0B5 to 0x0BB, checksum at 0x0BC, not checked
- Address 0x0BD to 0x0C3, checksum at 0x0C4, not checked
- Address 0x0C5 to 0x0CB, checksum at 0x0CC, not checked
- Address 0x0CD to 0x0D3, checksum at 0x0D4, not checked
- Address 0x0D5 to 0x0DB, checksum at 0x0DC, not checked
- Address 0x0DD to 0x0E3, checksum at 0x0E4, not checked
- Address 0x0E5 to 0x0EB, checksum at 0x0EC, not checked
- Address 0x0ED to 0x0F3, checksum at 0x0F4, not checked
- Address 0x0F5 to 0x0FB, checksum at 0x0FC, not checked
- Address 0x0FD to 0x103, checksum at 0x104, not checked

- Address 0x105 to 0x10B, checksum at 0x10C, not checked
- Address 0x10D to 0x113, checksum at 0x114, not checked
- Address 0x115 to 0x11B, checksum at 0x11C, not checked
- Address 0x11D to 0x123, checksum at 0x124, not checked
- Address 0x125 to 0x12B, checksum at 0x12C, not checked
- Address 0x12D to 0x133, checksum at 0x134, not checked
- Address 0x135 to 0x13B, checksum at 0x13C, not checked
- Address 0x13D to 0x143, checksum at 0x144, not checked
- Address 0x145 to 0x14B, checksum at 0x14C, not checked
- Address 0x14D to 0x153, checksum at 0x154, not checked
- Address 0x155 to 0x15B, checksum at 0x15C, not checked
- Address 0x15D to 0x163, checksum at 0x164, not checked
- Address 0x165 to 0x16B, checksum at 0x16C, not checked
- Address 0x16D to 0x173, checksum at 0x174, not checked
- Address 0x175 to 0x17B, checksum at 0x17C, not checked
- Address 0x17D to 0x183, checksum at 0x184, not checked
- Address 0x185 to 0x18B, checksum at 0x18C, not checked
- Address 0x18D to 0x1B2, no checksum
- Address 0x1B3 to 0x1DD, checksum at 0x1DE, Error #61

If you use the bit banger (EEPROM Edit Utility) in the RSS, you will find that the base address used is 0xB600. This is the equivalent to offset 0x0000 above.

If you use the Fix Checksum option (F6) the software will correct the checksum for first block listed (0xB600 to 0xB62D/0x000 to 0x02D).

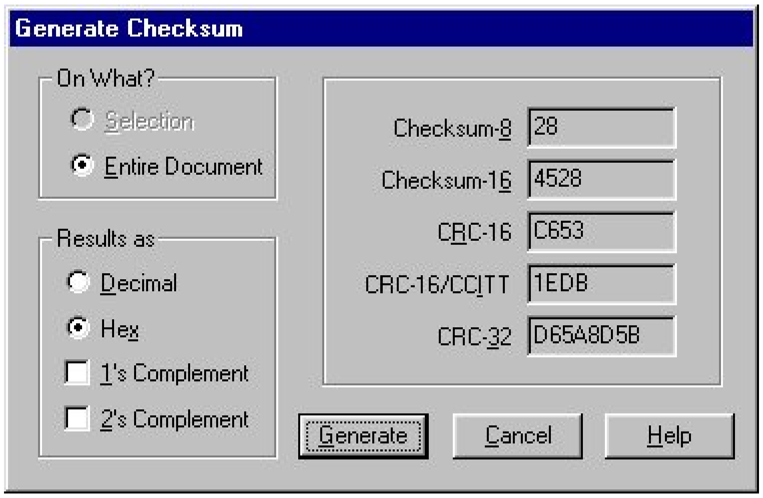
So, if you are hacking around in the codeplug (or changing the serial number), the locations of the checksums for each block will be important.

To calculate the checksum for a block of data, you will have to enter the data into Hex Workshop (version 2.54) manually. Once the block of data is enterred, highlight all the bytes except the checksum byte and hit F12 (Calculate Checksum). Make sure the Selection, Hex, and 1's Complement options are checked off, and click Generate. The value that is generated in the Checksum-8 field is what you want to use for the checksum for the block of data when you modify it in the RSS.

MAXTRAC: Try using the "Shift method" same as with the Radius and see if your version of RSS supports it. Otherwise, well, you will either need to do some Hex editing of your saved file or get a copy of the LAB RSS.

If you are going to use the Hex editing method, see the steps below.

In the MAXTRAC.MDF file make a Checksum-16 of the whole file (the F12 option in Hex Workshop), in our case we ended up with 0x4528 (RSS V 06.00), write this down, you might need it later.

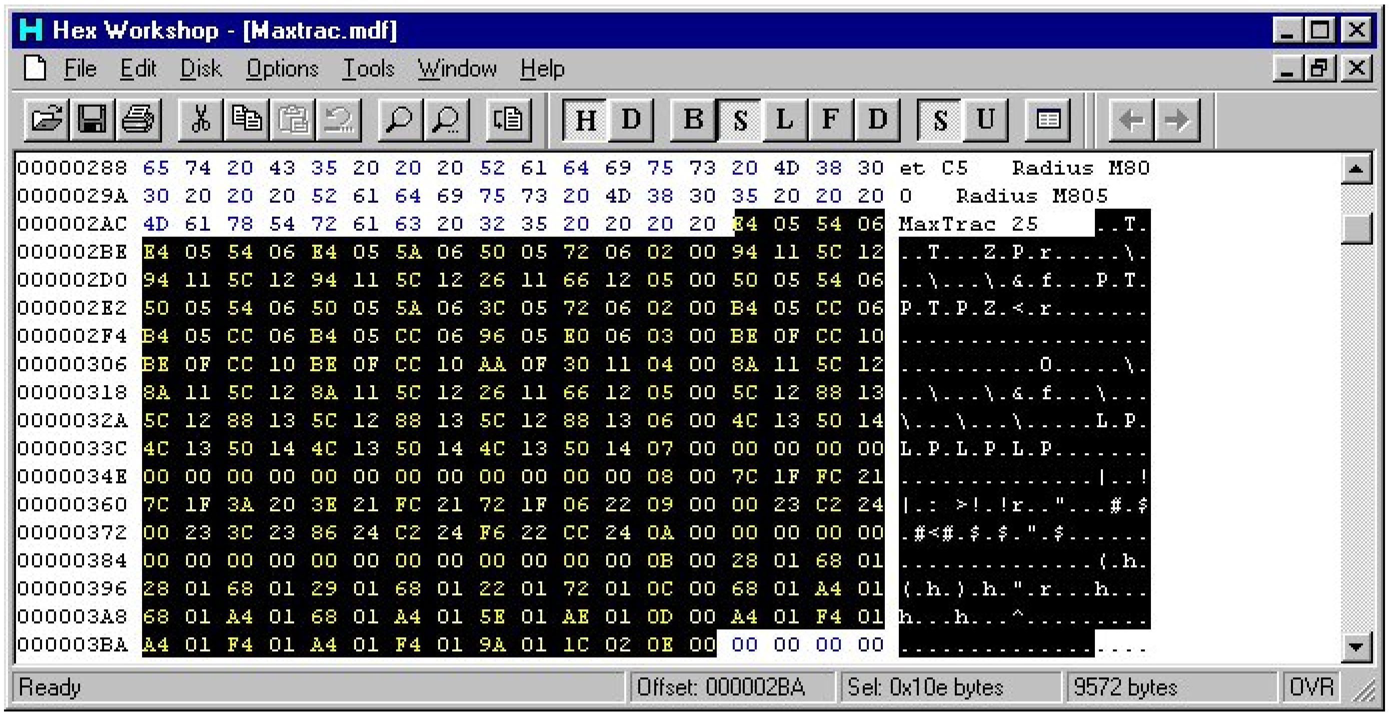


On What?: Selection, [✓ Entire Document]

Results as: Decimal, [✓ Hex], 1's Complement, 2's Complement

Checksum-8	28
Checksum-16	4528
CRC-16	C653
CRC-16/CCITT	1EDB
CRC-32	D65A8D5B

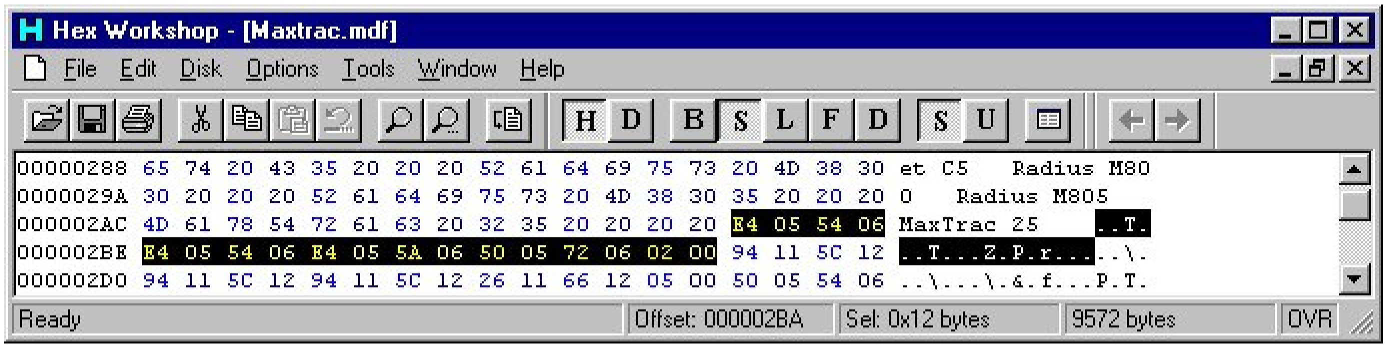
Look starting at about offset 0x2BA, you should see something like:



```
00000288 65 74 20 43 35 20 20 20 52 61 64 69 75 73 20 4D 38 30 et C5   Radius M80
0000029A 30 20 20 20 52 61 64 69 75 73 20 4D 38 30 35 20 20 20 0   Radius M805
000002AC 4D 61 78 54 72 61 63 20 32 35 20 20 20 20 E4 05 54 06 MaxTrac 25    ..T.
000002BE E4 05 54 06 E4 05 5A 06 50 05 72 06 02 00 94 11 5C 12 ..T...Z.P.r.....\.
000002D0 94 11 5C 12 94 11 5C 12 26 11 66 12 05 00 50 05 54 06 ..\...\.&.f...P.T.
000002E2 50 05 54 06 50 05 5A 06 3C 05 72 06 02 00 B4 05 CC 06 P.T.P.Z.<.r.......
000002F4 B4 05 CC 06 B4 05 CC 06 96 05 E0 06 03 00 BE 0F CC 10 ..................
00000306 BE 0F CC 10 BE 0F CC 10 AA 0F 30 11 04 00 8A 11 5C 12 ..........0.....\.
00000318 8A 11 5C 12 8A 11 5C 12 26 11 66 12 05 00 5C 12 88 13 ..\...\.&.f...\...
0000032A 5C 12 88 13 5C 12 88 13 5C 12 88 13 06 00 4C 13 50 14 \...\...\.....L.P.
0000033C 4C 13 50 14 4C 13 50 14 4C 13 50 14 07 00 00 00 00 00 L.P.L.P.L.P.......
0000034E 00 00 00 00 00 00 00 00 00 00 00 00 08 00 7C 1F FC 21 ..............|..!
00000360 7C 1F 3A 20 3E 21 FC 21 72 1F 06 22 09 00 00 23 C2 24 |.: >!.!r.."...#.$
00000372 00 23 3C 23 86 24 C2 24 F6 22 CC 24 0A 00 00 00 00 00 .#<#.$.$.".$......
00000384 00 00 00 00 00 00 00 00 00 00 00 00 0B 00 28 01 68 01 ..............(.h.
00000396 28 01 68 01 29 01 68 01 22 01 72 01 0C 00 68 01 A4 01 (.h.).h.".r...h...
000003A8 68 01 A4 01 68 01 A4 01 5E 01 AE 01 0D 00 A4 01 F4 01 h...h...^.........
000003BA A4 01 F4 01 A4 01 F4 01 9A 01 1C 02 0E 00 00 00 00 00 ..................
Ready                          Offset: 000002BA   Sel: 0x10e bytes   9572 bytes   OVR
```

The highlighted area contains all the bandsplits recognized by this particular RSS.

A single bandplit entry in the table looks like:



```
00000288 65 74 20 43 35 20 20 20 52 61 64 69 75 73 20 4D 38 30 et C5   Radius M80
0000029A 30 20 20 20 52 61 64 69 75 73 20 4D 38 30 35 20 20 20 0   Radius M805
000002AC 4D 61 78 54 72 61 63 20 32 35 20 20 20 20 E4 05 54 06 MaxTrac 25    ..T.
000002BE E4 05 54 06 E4 05 5A 06 50 05 72 06 02 00 94 11 5C 12 ..T...Z.P.r.....\.
000002D0 94 11 5C 12 94 11 5C 12 26 11 66 12 05 00 50 05 54 06 ..\...\.&.f...P.T.
Ready                          Offset: 000002BA   Sel: 0x12 bytes   9572 bytes   OVR
```

If you look carefully at the highlighted portion, you should notice the following items>

02 00 = Bandsplit identifier (Note: it is at the end of the frequency assignments)

E4 05 = 1508 in decimal, ie 150.8000 MHz

54 06 = 1620 in decimal, ie 162.0000 MHz

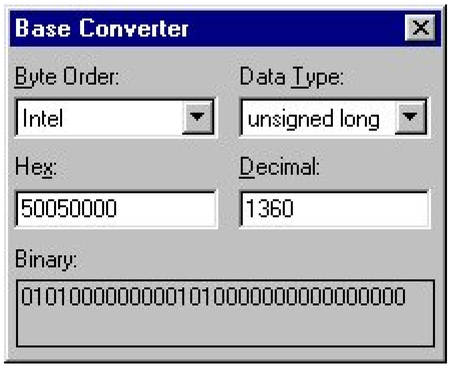
Note there are 3 occurances of the strings E4 05 and 54 06, and another set of limits, 50 05 and 72 06. The first set is the limits displayed in the Radio Wide menu (F4-F2). The second set are the TX limits for the radio. The third set are the RX limits for the radio (note that if the TX and RX limits are not set the same the RX limits are screwed up). The fourth set are the limits that the radio is spec'ed to go and that Lab RSS would let you program (only giving an "Outside of FCC Limits" error), in this case the limits translate to 136.0000 MHz to 165.0000 MHz.

If you program outside these limits the Lab RSS reports that the frequencies are "Outside of FCC and Radio Bandwidth Limits". In other words, if you want to have your radio operate properly, don't exceed the limits in the "Lab" field.

To figure out what hex to enter for your bandsplits you can either use the Base Converter (in Hex Workshop), or use a calculator with a HEX --> DECIMAL conversion function.

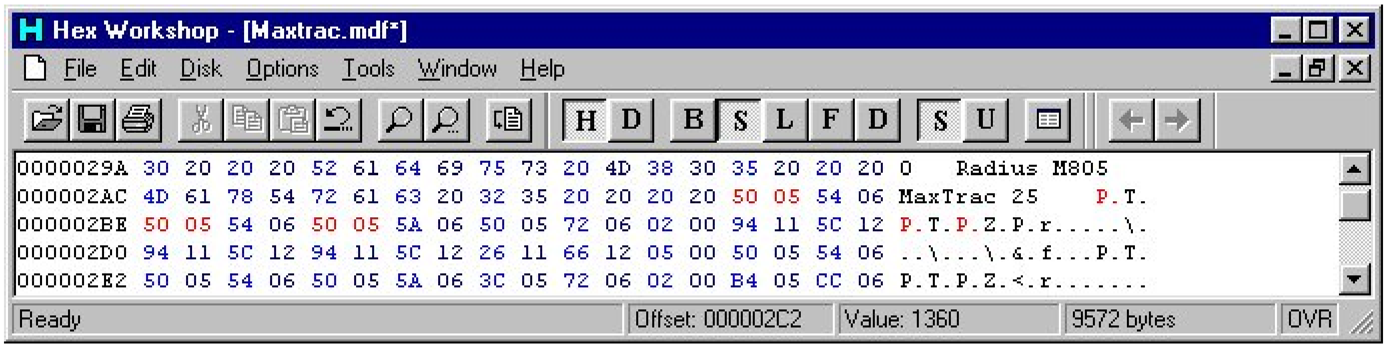
If you are using a calculator enter your limit (ie 1360 for 136.000 MHz) and convert it to hex, you should get 0x0550. When you enter the data into the .mdf file you have to reverse the bits such that you would actually enter 0x5005 in the field you are changing.

If you are using Hex Workshop, launch the Base Converter utility and select "Intel Byte Order". Then, enter your desired frequency and write down the hex result.



Byte Order: Intel
Data Type: unsigned long
Hex: 50050000
Decimal: 1360
Binary: 01010000000001010000000000000000

You can then directly enter the hex result (0x5005) into the bandsplit field.


Hex Workshop - [Maxtrac.mdf*]
File Edit Disk Options Tools Window Help
H D B S L F D S U

```
0000029A 30 20 20 20 52 61 64 69 75 73 20 4D 38 30 35 20 20 20 0   Radius M805
000002AC 4D 61 78 54 72 61 63 20 32 35 20 20 20 20 50 05 54 06 MaxTrac 25    P.T.
000002BE 50 05 54 06 50 05 5A 06 50 05 72 06 02 00 94 11 5C 12 P.T.P.Z.P.r.....\.
000002D0 94 11 5C 12 94 11 5C 12 26 11 66 12 05 00 50 05 54 06 ..\...\.&.f...P.T.
000002E2 50 05 54 06 50 05 5A 06 3C 05 72 06 02 00 B4 05 CC 06 P.T.P.Z.<.r.......
Ready | Offset: 000002C2 | Value: 1360 | 9572 bytes | OVR
```

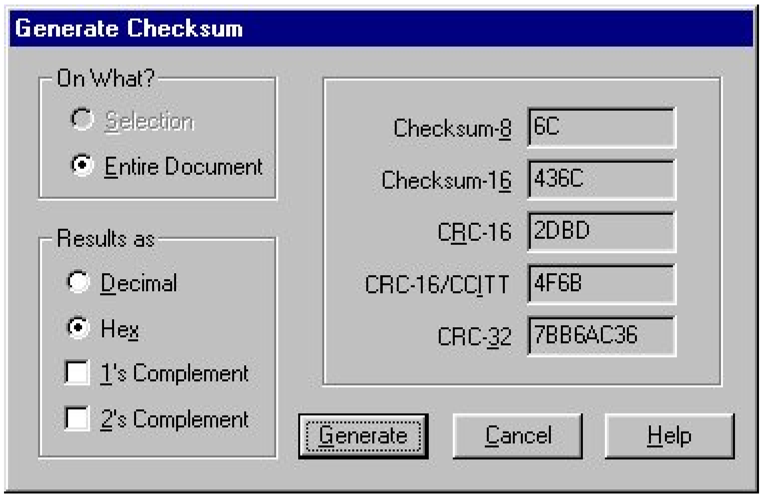
In this example we are changing the lower bandsplit of a 150.8-162MHz radio to 136-162MHz.

Save the new file (you might want to make a backup copy of the original if you haven't already.

Some of the RSS packages (we don't have codeplugs for everything, so we can't check them all) check the checksum of the .mdf file when you try and load a codeplug and return an error if the checksum of the .mdf file doesn't match the one stored in the program. Try running the RSS with the new .mdf file, if you don't get an error when you run the RSS and load a codeplug then you don't have to worry about the next step.

Correcting the .mdf File Checksum

If you get a corrupt .mdf file error then you will have to go back and correct the checksum in your edited .mdf file. Load the file back in your hex editor and make a Checksum-16 of the file. If you compare the new checksum

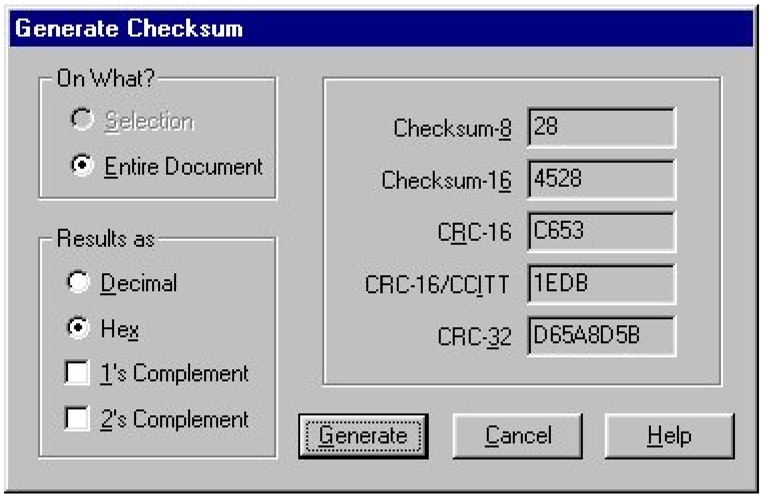


On What? ( ) Selection [✓ Entire Document]

Results as: ( ) Decimal [✓ Hex] [ ] 1's Complement [ ] 2's Complement

Checksum-8	6C
Checksum-16	436C
CRC-16	2DBD
CRC-16/CCITT	4F6B
CRC-32	7BB6AC36

with the original one



On What? ( ) Selection [✓ Entire Document]

Results as: ( ) Decimal [✓ Hex] [ ] 1's Complement [ ] 2's Complement

Checksum-8	28
Checksum-16	4528
CRC-16	C653
CRC-16/CCITT	1EDB
CRC-32	D65A8D5B

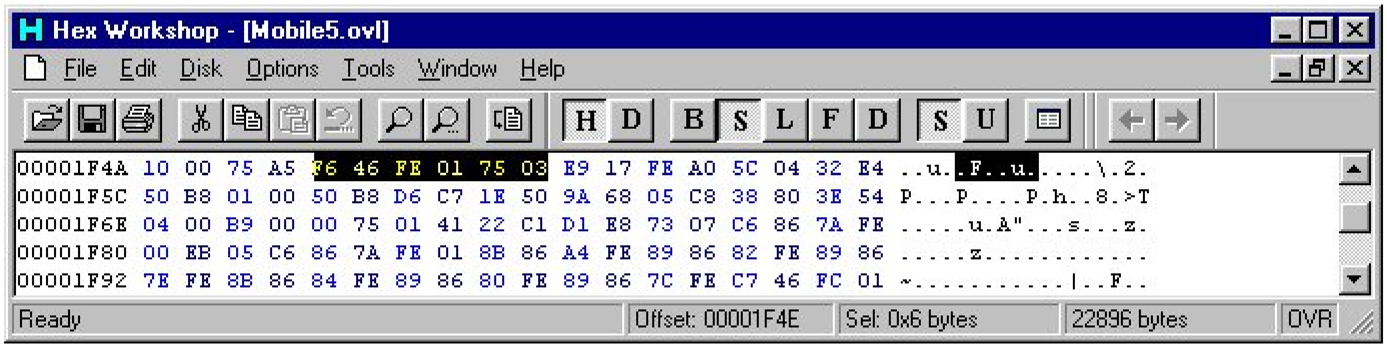
You will find they are probably different.

The only way we have to correct the checksum of the file at this time is to keep editing bits and making Checksum-16 calculations until the edited file's checksum and the original match. You can either edit the Copyright statement or some of the model descriptions in the file. Just take one or a few of these insignificant bytes (some experimentation may be required depending on by how much the cheksum is out) and add or subtract a few bits of the numbers make a Checksum-16 of the file. You should notice the checksum has changed by the number of bits you added or subtracted. Keep going until the checksum's match.

Once the checksum's of the files match you should be able to run the RSS, load the codeplug, and enter the frequencies within your new bandsplits with ease.

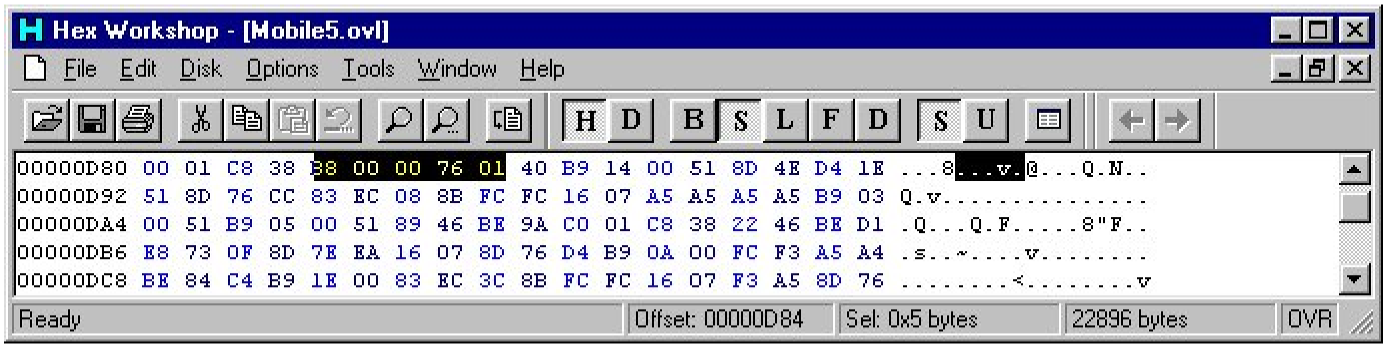
Changing Band Limits in Maxtrac RSS Version 5.03

Search for F6 46 FE 01 75 03 in file MOBILE5.OVL and change the 0xF6 to 0xC6.


Hex Workshop - [Mobile5.ovl]
File Edit Disk Options Tools Window Help

```
00001F4A 10 00 75 A5 F6 46 FE 01 75 03 E9 17 FE A0 5C 04 32 E4 ..u..F..u....\.2.
00001F5C 50 B8 01 00 50 B8 D6 C7 1E 50 9A 68 05 C8 38 80 3E 54 P...P....P.h..8.>T
00001F6E 04 00 B9 00 00 75 01 41 22 C1 D1 E8 73 07 C6 86 7A FE .....u.A"...s...z.
00001F80 00 EB 05 C6 86 7A FE 01 8B 86 A4 FE 89 86 82 FE 89 86 .....z............
00001F92 7E FE 8B 86 84 FE 89 86 80 FE 89 86 7C FE C7 46 FC 01 ~...........|..F..
Ready    Offset: 00001F4E    Sel: 0x6 bytes    22896 bytes    OVR
```

Search for B8 00 00 76 01 and change the 0x76 to 0xEB.


Hex Workshop - [Mobile5.ovl]
File Edit Disk Options Tools Window Help

```
00000D80 00 01 C8 38 B8 00 00 76 01 40 B9 14 00 51 8D 4E D4 1E ...8...v.@...Q.N..
00000D92 51 8D 76 CC 83 EC 08 8B FC FC 16 07 A5 A5 A5 A5 B9 03 Q.v...............
00000DA4 00 51 B9 05 00 51 89 46 BE 9A C0 01 C8 38 22 46 BE D1 .Q...Q.F.....8"F..
00000DB6 E8 73 0F 8D 7E EA 16 07 8D 76 D4 B9 0A 00 FC F3 A5 A4 .s..~....v........
00000DC8 BE 84 C4 B9 1E 00 83 EC 3C 8B FC FC 16 07 F3 A5 8D 76 .........<.......v
Ready    Offset: 00000D84    Sel: 0x5 bytes    22896 bytes    OVR
```

You can now program any frequency in, it is a matter if the hardware will do it.

Adding Type II/IIi Trunking to a Type I Maxtrac

First off you must start with a B5 B6 or B7 model (has to be 16 pin ACC)

The second step is to obtain a type II firmware chip for a Smartnet Maxtrac, the part number is HLN9383.

Next, you need to blank the logic board using Lab RSS.

Re-program the logic board with your trunking RSS so that it is now a Smartnet radio (either a C3, C5, or C6).

Realign the unit and you now have a unit capable of type I,II, and IIi.

Making a Maxtrac do PAC-RT

One similar way this can be economically achieved is by using a RICK unit, or building a repeater cable (see at the beginning of this page) yourself and plugging it into a second MaxTrac.

The second MaxTrac can be a simple 2-channel unit (or not) and will two-way repeat the first one and vice-versa. You can also construct this device to one-way repeat (just eliminating the second TX line) to construct a GMRS repeater out of two Maxtracs.

Otherwise, to actually use a PAC-RT with a Maxtrac you need the following cable:

```
MaxTrac                                 PAC-RT
Acc Conn.
        3  -------------------- 6
        13 -------------------- 1
        11 -------------------- 21 (Shielded Cable)
        2  -------------------- 19 (Shielded Cable)
        7  -------------------- 8
                          ----- 2
                          |
                          ----- 16
                          ----- 10
                          |
                          ----- 17
```

Cable shields are connected to PAC-RT Pin 18 (Ground) and not connected at radio. There are two jumpers inside the PAC-RT Connector, Pin 2 to 16 (B+) and 10 to 17 (PTT Ground).

```
For on-off control:
PAC-RT  Pin 2 B+ to on/off switch
        Pin 9 & 15 other side of on/off switch
(Pin 9 = PAC on/off, Pin 15 = Portable in/out)
Both must be connected to B+ for unit to turn on,
Pin 22 ground (not required, unless you want an indicator lamp, or something).
```

---

99 Channel Modification

Yes, there is a 99 channel Maxtrac. It was a SP model, low band unit that was made for Ontario Hydro in Eastern Canada. The model number is ACD51MJA9HA5AK and covers 42-50 MHz.

This radio was basically a stock low band radio, with a modified HLN9313A logic board. The logic board has a larger EEPROM installed to hold the very large codeplug as well as different (VLN5443A) firmware.

This modified radio is only able to be programmed with VVN4168A RSS. This is SP Maxtrac RSS that only supports this model.

If you are interested in the original manual supplement that covers this model, click to see page 1 and page 2.

You can convert a normal Maxtrac with the HLN9313A, 16 pin logic board, to a 99 channel model by following the instructions contained in the manual supplement. Once you expand the EEPROM and put the new firmware in, you will need to blank the logic board with Maxtrac Lab RSS and then re-initialize it with the 99 channel RSS service menu. Once you do that, you should have a radio that is capable of 99 channels, you will need to hack the .mdf file depending on what the bandsplit of the RF board is in your radio in order to be able to program the proper frequencies into it.

FYI, the firmware and RSS is NLA from Motorola. Good luck in locating it.

Panel Numbers for Lab RSS

OK, so you're hacking your poor Maxtrac with Lab RSS and have blanked the board. You are in the process of re-initializing the logic board using the Replace Logic Board options in the Service Menu. BUT, you've hit a snag... the RSS is asking you for something called a "Panel Number". What the heck is that?! Well, look no further. If you had been wise and used the built-in help (F1) when you got to this point, you would probably have found the following list, but you didn't, and now you are here.

Here are the panel numbers used when re-initializing various flavors of Maxtrac:

Panel Number	Description
0	All 2 frequency models: Radius M100, Maxtrac50, Maxtrac100; Voice Reporter Models: H1135_, H1136_, H1138_, H1139_, H1134_
1	All 6, 16 and 32 Frequency Models: Maxtrac 300

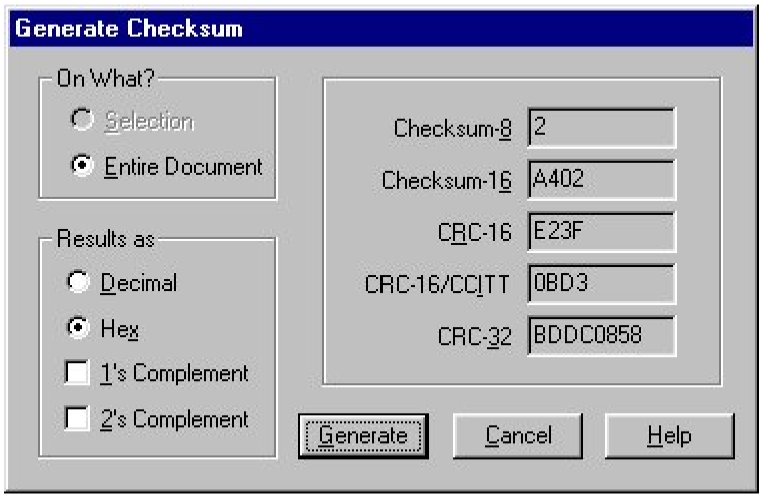
2	All 6 Frequency Models with Signalling or Accessory Connector Option: Maxtrac 300
50	Maxtrac 820 Trunked: D__MQA5GB5_K, D__MQA5GB1_K; Privacy Plus 250/750: T_5CPA5___H
51	Maxtrac 820 Trunked: D__MQA5GB3_K
52	Maxtrac 840 Trunked: D__MWA5GB5_K, D__MWA5GB6_K
53	Maxtrac 840 Trunked: D__MWA5GB7_K
54	Maxtrac 840 Trunked: D__MWA5GC0_K
55	Trunked Voice Reporter: H5011_
56	Maxtrac 820 Trunked: D__MQA5GB4_K

RADIUS:

Life can be a lot easier for Radius owners. All you have to do is load your file into RSS, go the channel you want to program, when you are in the field you want to change, use the shift key and the numbers across the top of the keyboard to enter the frequency. Fill the entire field and make sure you release the shift key when you enter the decimal point. ie. 139.5500 would be !#(.%%)) Its as simple as that.

If the above method doesn't work, then you'll have to do some hex editing.

In the RADMBL.MDF file make a Checksum-16 of the whole file (the F12 option in Hex Workshop), in our case we ended up with 0xA402 (RSS V 08.00), write this down, you might need it later.


Generate Checksum

On What? Selection [✓ Entire Document]

Results as Decimal [✓ Hex]

1's Complement ☐

2's Complement ☐

Checksum-8: 2

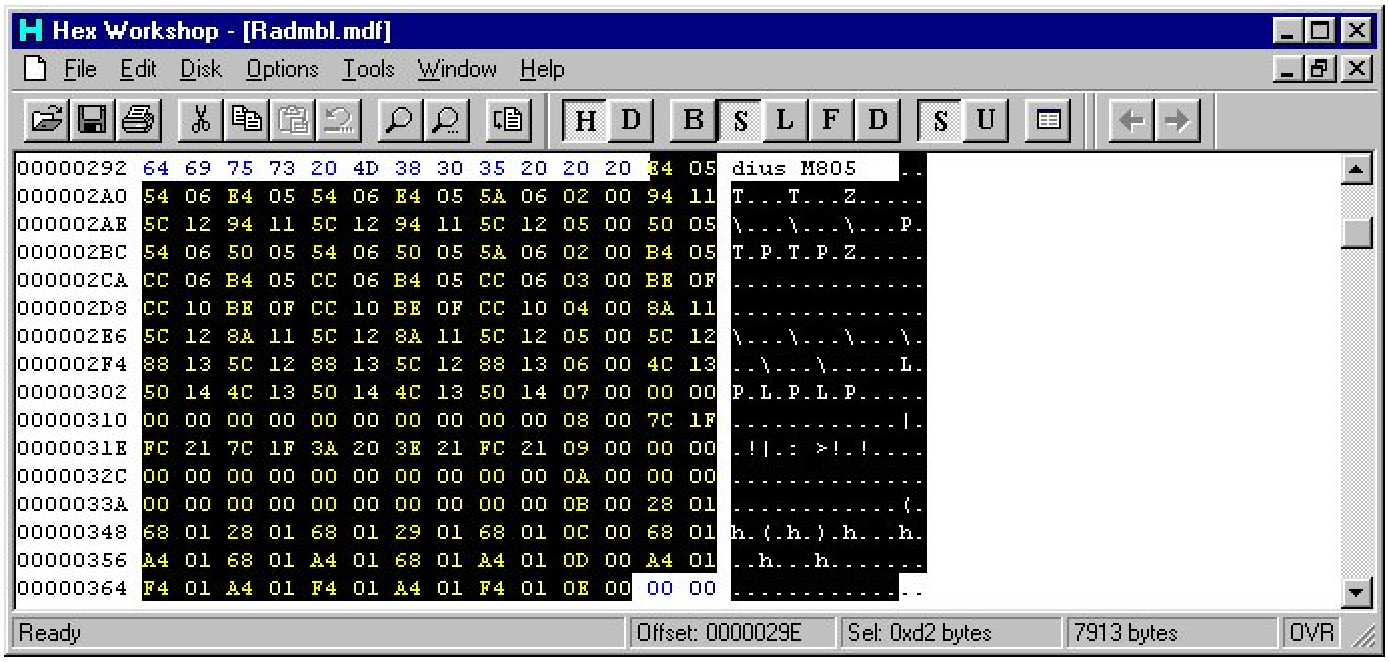
Checksum-16: A402

CRC-16: E23F

CRC-16/CCITT: 0BD3

CRC-32: BDDC0858

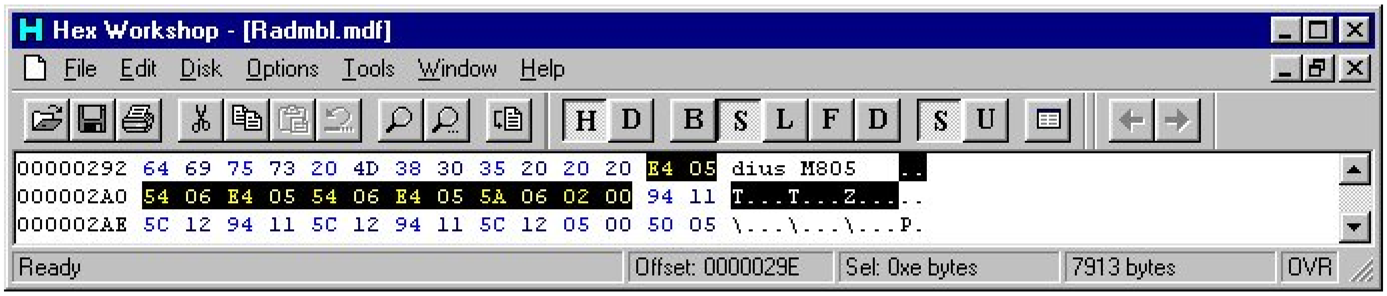
Look starting at about offset 0x29E, you should see something like:


Hex Workshop - [Radmbl.mdf]
File Edit Disk Options Tools Window Help

```
00000292 64 69 75 73 20 4D 38 30 35 20 20 20 E4 05 dius M805   ..
000002A0 54 06 E4 05 54 06 E4 05 5A 06 02 00 94 11 T...T...Z.....
000002AE 5C 12 94 11 5C 12 94 11 5C 12 05 00 50 05 \...\...\...P.
000002BC 54 06 50 05 54 06 50 05 5A 06 02 00 B4 05 T.P.T.P.Z.....
000002CA CC 06 B4 05 CC 06 B4 05 CC 06 03 00 BE 0F ..............
000002D8 CC 10 BE 0F CC 10 BE 0F CC 10 04 00 8A 11 ..............
000002E6 5C 12 8A 11 5C 12 8A 11 5C 12 05 00 5C 12 \...\...\...\.
000002F4 88 13 5C 12 88 13 5C 12 88 13 06 00 4C 13 ..\...\.....L.
00000302 50 14 4C 13 50 14 4C 13 50 14 07 00 00 00 P.L.P.L.P.....
00000310 00 00 00 00 00 00 00 00 00 00 08 00 7C 1F ............|.
0000031E FC 21 7C 1F 3A 20 3E 21 FC 21 09 00 00 00 .!|.: >!.!....
0000032C 00 00 00 00 00 00 00 00 00 00 0A 00 00 00 ..............
0000033A 00 00 00 00 00 00 00 00 00 00 0B 00 28 01 ............(.
00000348 68 01 28 01 68 01 29 01 68 01 0C 00 68 01 h.(.h.).h...h.
00000356 A4 01 68 01 A4 01 68 01 A4 01 0D 00 A4 01 ..h...h.......
00000364 F4 01 A4 01 F4 01 A4 01 F4 01 0E 00 00 00 ..............
Ready    Offset: 0000029E    Sel: 0xd2 bytes    7913 bytes    OVR
```

The highlighted area contains all the bandsplits recognized by this particular RSS.

A single bandplit entry in the table looks like:


Hex Workshop - [Radmbl.mdf]
File Edit Disk Options Tools Window Help

```
00000292 64 69 75 73 20 4D 38 30 35 20 20 20 E4 05 dius M805   ..
000002A0 54 06 E4 05 54 06 E4 05 5A 06 02 00 94 11 T...T...Z.....
000002AE 5C 12 94 11 5C 12 94 11 5C 12 05 00 50 05 \...\...\...P.
Ready    Offset: 0000029E    Sel: 0xe bytes    7913 bytes    OVR
```

If you look carefully at the highlighted portion, you should notice the following items>

02 00 = Bandsplit identifier (Note: it is at the end of the frequency assignments)

E4 05 = 1508 in decimal, ie 150.8000 MHz

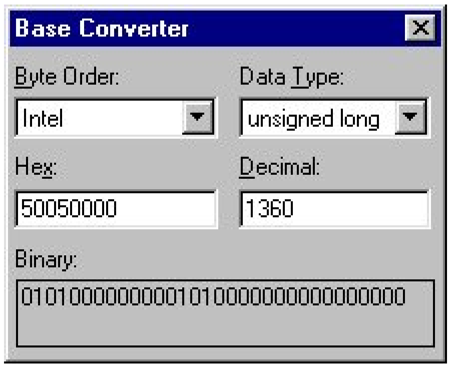
54 06 = 1620 in decimal, ie 162.0000 MHz

Note there are 3 occurances of the strings E4 05 and 54 06. The first set is the limits displayed in the Radio Wide menu (F4-F2). The second set are the TX limits for the radio. The third set are the RX limits for the radio (note that if the TX and RX limits are not set the same the RX limits are screwed up).

To figure out what hex to enter for your bandsplits you can either use the Base Converter (in Hex Workshop), or use a calculator with a HEX --> DECIMAL conversion function.

If you are using a calculator enter your limit (ie 1360 for 136.000 MHz) and convert it to hex, you should get 0x0550. When you enter the data into the .mdf file you have to reverse the bits such that you would actually enter 0x5005 in the field you are changing.

If you are using Hex Workshop, launch the Base Converter utility and select "Intel Byte Order". Then, enter your desired frequency and write down the hex result.


Base Converter

Byte Order: Intel

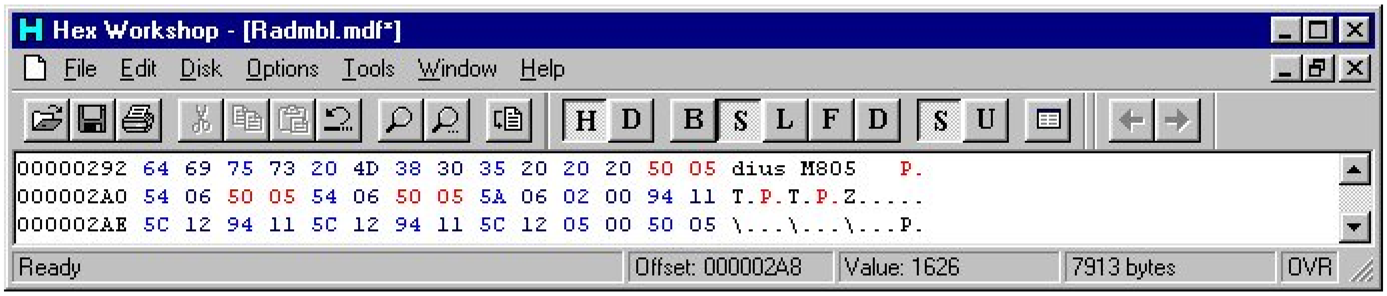
Data Type: unsigned long

Hex: 50050000

Decimal: 1360

Binary: 01010000000001010000000000000000

You can then directly enter the hex result (0x5005) into the bandsplit field.


Hex Workshop - [Radmbl.mdf*]

```
00000292 64 69 75 73 20 4D 38 30 35 20 20 20 50 05 dius M805    P.
000002A0 54 06 50 05 54 06 50 05 5A 06 02 00 94 11 T.P.T.P.Z.....
000002AE 5C 12 94 11 5C 12 94 11 5C 12 05 00 50 05 \...\...\...P.
```

Ready | Offset: 000002A8 | Value: 1626 | 7913 bytes | OVR

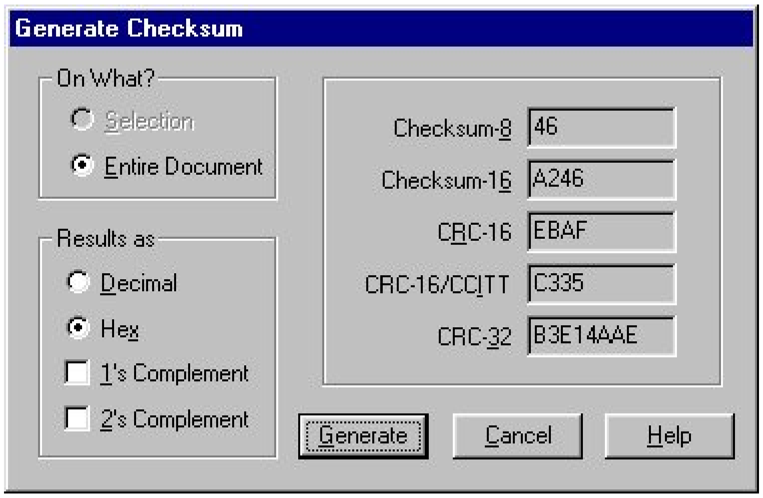
In this example we are changing the lower bandsplit of a 150.8-162MHz radio to 136-162MHz.

Save the new file (you might want to make a backup copy of the original if you haven't already.

Some of the RSS packages (we don't have codeplugs for everything, so we can't check them all) check the checksum of the .mdf file when you try and load a codeplug and return an error if the checksum of the .mdf file doesn't match the one stored in the program. Try running the RSS with the new .mdf file, if you don't get an error when you run the RSS and load a codeplug then you don't have to worry about the next step.

Correcting the .mdf File Checksum

If you get a corrupt .mdf file error then you will have to go back and correct the checksum in your edited .mdf file. Load the file back in your hex editor and make a Checksum-16 of the file. If you compare the new checksum


Generate Checksum

On What? Selection [✓ Entire Document]

Results as Decimal [✓ Hex] 1's Complement 2's Complement

Checksum-8: 46

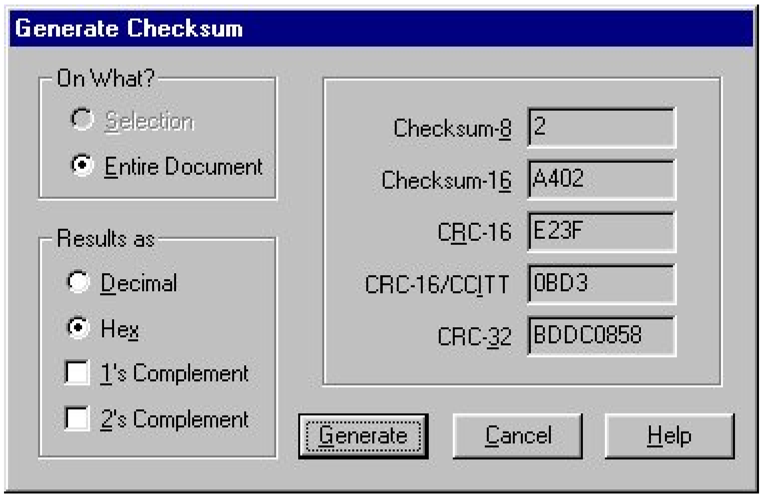
Checksum-16: A246

CRC-16: EBAF

CRC-16/CCITT: C335

CRC-32: B3E14AAE

with the original one


Generate Checksum

On What?
Selection
[✓ Entire Document]

Results as
Decimal
[✓ Hex]
1's Complement
2's Complement

Checksum-8	2
Checksum-16	A402
CRC-16	E23F
CRC-16/CCITT	0BD3
CRC-32	BDDC0858

You will find they are probably different.

The only way we have to correct the checksum of the file at this time is to keep editing bits and making Checksum-16 calculations until the edited file's checksum and the original match. You can either edit the Copyright statement or some of the model descriptions in the file. Just take one or a few of these insignificant bytes (some experimentation may be required depending on by how much the cheksum is out) and add or subtract a few bits of the numbers make a Checksum-16 of the file. You should notice the checksum has changed by the number of bits you added or subtracted. Keep going until the checksum's match.

Once the checksum's of the files match you should be able to run the RSS, load the codeplug, and enter the frequencies within your new bandsplits with ease.

Home