

So color by color you'll understand each part:

```
;K790 SW-R8BF003
; Example VKP Patch
;(i) Heap shift (201A53A4 - 201A53A8)
;Â© Sic
;(p) Se-MaG
+44140000
```

That's the info of the patch, each line has a **;** at the beginning so it doesn't change or add anything to the Main....

```
->The first line in the patch must include the phone model and the Firmware the
patch is for....
```

->After that line is the info of the patch, in here you specify what changes or adds makes the patch to the Main, like ability to use Camera when java is in background, and so on...

->After that line goes the important information after a (i) or (!) in her you write incompatibility to other patches, or if it's needed to apply some patch to make it work, in this case the Heapshift patch...(i) Heap shift)

->After you should put the credits to the Creator of the patch, (c) Shadow Player
it means the patch was created by Shadow and all the ports of the patch must
include that line...

->After you put the porting information, (p) Edgpaez it means the patch created by Shadow was ported by a different Firmware by Edgpaez

```
->After you may put (r) and the name of the person or web you got a lot of help for
creating/porting the patch.
```

+44140000

Is the base of the Firmware, it depends on each model, ALL DB2020 phones have 44140000 base address, as well as some db2010 phones.....

Sometimes patches doesn't have this line, because the base has already been added to the offsets.

```
c3d3a0: 211C A847
c3d580: 580D0000 01AFB045
deae2: 2418 8047
deeb08: 3C080000 11AFB045[/color]
def21a: 82200001201884B06A46 0148804701E059AFB045
def28c: C10F0000 BF0F0000
c47bee: 201C00F0D2FA 2F4880470000
c47ca6: 012000BD0000 F9E73BAFB045
c4a8e4: 059F 9847
c4a938: C0920420 89AFB045[/color]
```

In this and the last part there are 3 parts of code:

```
-> c3d3a0: This is called offset, it is the adress of the main where you can find
the bytes that will be changed or added, after it it must be added :
```

->211C: These are the Original bytes you can find in your main if you go to the offset before it.

```
->A847: There are the New bytes that will replace the original bytes in that offset
in order to change/add functions to the Main
```

So what this parts do when is applied to the phone is changing several bytes in the Main according to the offsets....

[illegible]

19caf10:	000000000000000000000000000000000000	24482418FFB5002100F03CF82DE0FFFF
19caf20:	000000000000000000000000000000000000	FFB5012100F036F800201F4FB8477D20
19caf30:	000000000000000000000000000000000000	C00012A101311A4F1EE0FFB51B481C4F
19caf40:	000000000000000000000000000000000000	B847002805D1184FB847012100F022F8
19caf50:	000000000000000000000000000000000000	13E00021174F0FE041204001201884B0
19caf60:	000000000000000000000000000000000000	6A46FFB5144FB847002806D030681349
19caf70:	000000000000000000000000000000000000	02A20132124B134FB847FFBDBFFB51248
19caf80:	000000000000000000000000000000000000	0068124FB847E4E7114B059FFFB50E4B
19caf90:	000000000000000000000000000000000000	1C60F2E7FFB50F480F4FEDE7580D0000
19cafa0:	000000000000000000000000000000000000	B97826453C0800009981D8440DEBF244
19cafb0:	000000000000000000000000000000000000	C5912645B1AD2645E5B6D844AB0F0000
19caf0c0:	000000000000000000000000000000000000	2C06000045FDF244A4531A2091912645
19cafd0:	000000000000000000000000000000000000	8C8704202AF40000F9FC2745

In this part we also have offsets, Original and New bytes... and it's the same than for the last part of code, the main difference here is that this part of code doesn't change anything to the original Main, what it does is add new functions and instructions to the Main.

At the end of the DB2020 Main files there are MANY 0000 (empty space) where this new code is added. in the DB2010 phones this new code is added in the space between the Main and the FS, so it changes (FFFF) for the new code.

Hopefully this little explanation gives you a great start to your way as a Developer, so feel free to post any of your doubts about VKP patches in here :P

Best wishes,
Se Developers