

# Universe Modding in X3 Terran Conflict (doc v1.03)

## Initial Deconstruction of "x3\_universe.xml"

To begin with - everything in the xml file is an "Object", I'm guessing that's what they had in mind when they named the xml Element "<o>".

What determines the type of object is the 't' attribute, for instance t="1" is a sector and t="17" is an asteroid.

So initial deconstruction seems to imply that

```
<o id="29" t="1" ... ></o>
```

... is a sector (type 1) with an id of 29

Let's look at Argon Prime...

```
<o id="49" f="4" t="1" x="1" y="3" r="1" size="22500000" m="8100" p="189451753" qtrade="100" qfight="100" qbuild="100" qthink="100">
```

### This is an deconstruction of the Argon Prime Object Element:

<b>id = "49"</b>	this is the object's unique id
<b>f="4"</b>	f is a special flag which, in the case of sectors, only has a value of 4 and denotes a Core sector
<b>x="1"</b>	is the 'x' index of the sector - this is based on a zero index so 1 is actually the second row of sectors from the left
<b>y="3"</b>	is the 'y' index of the sector - again this is based on a zero index, but y is "down" rather than up - sector 0, 0 for instance is "Kingdoms End" and 1, 3 is Argon Prime
<b>r="1"</b>	is the 'race' – see appendix 2
<b>size="22500000"</b>	this is to do with the gate distance on the map, it seems to be a measure of radius... the gate distance is, however a diameter so you need to multiply by 2. This gives you $2 * (22500000 / 500000) = 90\text{km}$ BINGO! However, when displayed this is always rounded to the nearest 5km.
<b>m="8100"</b>	this is the music track - Argon prime is 08100.mp3 - 0 seems to denote using a default track
<b>p="189451753"</b>	this is the population, a figure of -1 seems to denote using a default value, however I don't know where this comes from. A figure of -2 will display "unknown" (used in "unknown sectors")
<b>qtrade="100"</b>	I think this denotes the frequency of "trade" missions (the 'q' stands for quota)
<b>qfight="100"</b>	ditto for "fight" missions
<b>qbuild="100"</b>	ditto for "build" missions
<b>qthink="100"</b>	ditto for "think" missions

## Types

We've already noted that types are defined by the 't' attribute and that t="1" is a sector. Following is an investigation into the various "object" types:

### T1: Sectors

See "Initial Deconstruction" (above) for an explanation of Sectors.

### T2: (Sector) Background

There are 4 attributes here:

<b>t</b>	is the "type" and is always 2
<b>s</b>	is a sub-type which refers to the index (line number counting from 0) in types\TBackgrounds.pck
<b>neb</b>	presumably a (currently) unused setting for nebula – I'm saying unused as it always has a value of 0 – however, there used to be different nebula types, sensor obscuring, shield damaging and so on and I'd assume this value tied into those indexes once upon a time.
<b>stars</b>	another unused setting; this is always 0

Note: a pck file is just a packed file... for all of the *types\TType.pck* files the file that has been packed is just a plain text (txt) file.

### T3: Suns

Suns have 6 attributes; the x, y and z values are the three dimensional co-ordinates that denote the position of the sun. These figures are generally pretty enormous so that the sun (light source) is positioned outside the navigable area of the sector.

There are 6 attributes here:

<b>t</b>	is the "type" and is always 3
<b>s</b>	is a sub-type which refers to the index in types\TSuns.pck
<b>x</b>	is the 'x' position of the sun (light source)
<b>y</b>	is the 'y' position of the sun
<b>z</b>	is the 'z' position of the sun
<b>color</b>	is an RGB colour index – see appendix 2

The "types\TSuns.pck" resides in 03.cat and is a comma separated values file (ok, technically semi-colon separated values file) so if you extract it and rename it to "TSuns.csv" you should be able to open it in Excel or OpenOffice Calc.

To Change the brightness of a sun type you need to tweak the value in the 'J' column, by default this will be either **65536** (100%) or **98304** (150%) [since  $98305/65536 = 1.5$ ]. So changing this value to 131072 **should** give you a sun brightness of 200%, or you could just add more suns to the sector to increase the brightness.

**Note:** the first line of any *TType* file is an informational marker – if you add a new line to this file you'll need to increment the line counter (11 in this case) – I think the other part refers to the generating script engine 41 for Reunion and 47 for Terran Conflict. Also, unless you're bundling it into a mod, odds are any tweaks in TType files will be overwritten when the game is next patched – **be warned!**

## T4: Planets

Planets again have type, sub-type and co-ordinate attributes, but they also have an **fn** attribute which I've not worked out yet.

There are 8 possible attributes here:

<b>f</b>	seems to denote a "special flag"; there are 17 planets where <b>f</b> ="1" but I'm not sure what that implies
<b>t</b>	is the "type" and is always 4
<b>s</b>	is a sub-type which refers to the index in types\TPlanets.pck
<b>x</b>	is the 'x' position of the planet
<b>y</b>	is the 'y' position of the planet
<b>z</b>	is the 'z' position of the planet
<b>color</b>	is an RGB colour index – see colours on page 2
<b>fn</b>	is apparently to do with "fogging" - I suspect it's defunct since the only planet that has it set is in Pontifex's Realm

**Note:** the number of planets a sector displays on the galaxy map does not include planets that have an **s** attribute of 128. There is no 128 index in TPlanets so I guess these default to the planet type SS\_PLDUMMY.

## T5: [Stations] Trading Docks

Type 5 stations are trading docks, equipment docks, military, supply and pirate bases. The wares available to these docks are defined not only from their prototype in the *WareTemplate.xml* file but also inside a custom container object (see T23) which extends the basic wares list.

Certain trading docks can also appear as wares in shipyards – these will have only the **t** and **s** attributes.

There are 11 possible attributes here:

<b>id</b>	is the unique ID for the station
<b>f</b>	denotes a "special flag", in the case of stations I think it has something to do with the scripts it's running
<b>t</b>	is the "type" and is always 5
<b>s</b>	is a sub-type which refers to the index in types\TDocks.pck – all of the important factory information (such as production RelVals and Price Modifiers exist in that file) – but not the wares list.
<b>x</b>	is the 'x' position of the trading dock
<b>y</b>	is the 'y' position of the trading dock
<b>z</b>	is the 'z' position of the trading dock
<b>r</b>	is the race id (see appendix 2) – any station can belong to any race.
<b>a</b>	
<b>b</b>	a, b and g are rotational factors – see appendix 1.
<b>g</b>	

**n** denotes a special dock variant (excluding the Saturn Research Station)  
3761 => Pirate Anarchy Port  
5071 => Royal Boron Research Station  
5061 => Federal Argon Installation  
5081 => Paranid Communications Facility

## T6: [Stations] Factories/Shipyards

Type 6 stations are “everything else” - basic materials factories, shipyards, space fuel - you name it, if it's a factory and it's not a trading/equipment dock, it's a type 5. Again a basic prototype wares list exists in *WareTemplate.xml* and again some factories, shipyards only by default, make use of the T23 custom container object.

All factories (except shipyards) can also appear as wares at shipyards. These factories will have only the **t** and **s** attributes.

There are 10 possible attributes here:

**id** is the unique ID for the station

**f** denotes a “special flag”, in the case of stations I think it has something to do with the scripts it's running

**t** is the “type” and is always 6

**s** is a sub-type which refers to the index in `types\TFactories.pck` – all of the important factory information (such as production `RelVals` and `Price Modifiers` exist in that file) – but not the wares list.

**x** is the 'x' position of the factory

**y** is the 'y' position of the factory

**z** is the 'z' position of the factory

**r** is the race id (see appendix 2) – any station can belong to any race.

**a**

**b** are rotational factors – see appendix 1.

**g**

## T7: Ships

All ships are of type 7 – this includes Orbital Weapons Platforms.

Ships can appear in three different ways in the `x3_universe.xml` file:

- In space – which generally refers to Orbital Weapons Platforms, unclaimed ships, mines and some special NavSats. These ships will have **id**, **x**, **y** and **z** attributes.
- Docked – OWP's seem to start with a small complement of fighters. These ships will have **ids**, but **not** **x**, **y** or **z** attributes. They will also be contained within a special T23 container object inside the OWP and may well have a T23 of their own.
- As a ware – shipyards have a T23 container which holds a list of their stock; in this case ships. These ships will **not** have **id**, **x**, **y**, or **z** attributes.

There are 8 possible attributes here:

<b>id</b>	is the unique ID for the ship (except for ships that are wares)
<b>f</b>	denotes a “special flag” - there are 6 OWPs with f=“1” and I think it denotes that a script is running, probably “attack all enemies” in the case of OWPs
<b>t</b>	is the “type” and is always 7
<b>s</b>	is a sub-type which refers to the index in types\TShips.pck – information such as speed and acceleration is stored in this file
<b>x</b>	is the 'x' position of the ship (if it's “in space”)
<b>y</b>	is the 'y' position of the ship (if it's “in space”)
<b>z</b>	is the 'z' position of the ship (if it's “in space”)
<b>r</b>	is the race id (see appendix 2) – any ship can belong to any race.

## T8-16 Wares

Wares that are part of the standard production of a station do not appear in the x3\_universe.xml file – for instance cahoonas bakeries don't contain meatsteak cahoonas as a ware, in fact they don't contain any wares at all the factory type itself determines its wares and resources by loading the data from the *WareTemplate.xml* file.

The galaxy editor allows you to add additional wares of any type, to any station and you can do this in the xml file too, for more information see T23 customisable containers.

The vast majority of wares in the universe file appear inside T23 containers, i.e. wares that are available to buy or sell at trading stations or equipment docks (and the like) – therefore they only have t and s attributes. However, if you want to put as a container floating in space, you can do so by adding a unique id, x, y and z coordinates and an n attribute (which defines how many).

For instance, to add 15 poltergeist missiles to a sector (and this is taken directly from the xml file so it actually *does* exist in a sector in the game) you can do the following:

```
<o id="7291" t="10" s="SS_MISSILE_POLTERGEIST" x="13292500" y="-4520387" z="9894669" n="15"/>
```

The different wares types are as follows:

- T8: Lasers**
- T9: Shields**
- T10: Missiles**
- T11: Energy**
- T12: Novelty items (things like insecticide)**
- T13: Basic resources such as argnu beef, sunrise flowers and delaxian wheat**
- T14: Consumables such as food, space fuel and cloth rimes**
- T15: Minerals (ore, silicon, nividium and ice)**
- T16: Tech – which incidentally includes slaves and police licenses**

There are 8 possible attributes here:

<b>id</b>	is the unique ID for the ware – this is only used when the ware is floating in space.
<b>f</b>	denotes a “special flag” - probably relates to a script. There is only 1 ware with this flag (f=“1”) in the universe map; a bunch of 5 Terran Keris drones you can pick up.
<b>t</b>	is the “type” - see above

- s** is the sub-type; the technical definitions of these sub-types exist in their relevant *TFiles* (based on their *t* attribute) – which *TFile* the ware can be found in depends on its type:
- T8: Lasers: types\TLaser.pck
  - T9: Shields: types\TShields.pck
  - T10: Missiles: types\TMissiles.pck
  - T11: Energy: types\TWareE.pck
  - T12: Novelties: types\TWareN.pck
  - T13: Basic: types\TWareB.pck
  - T14: Consumables: types\TWareF.pck (think food)
  - T15: Minerals: types\TWareM.pck
  - T16: Tech: types\TWareT.pck
- i** refers to the laser slot in which any weapon is installed (presumably missiles in the case of M7Ms as well). This is just an index number so if you have the ship info onscreen in front of you as you count down the weapons slots (counting from 0) that will be the index number. So on a Nova Vanguard 0-7 will be the front lasers and 8 will be the rear turret.
- x** is the 'x' position of the ware (if it's "in space")
- y** is the 'y' position of the ware (if it's "in space")
- z** is the 'z' position of the ware (if it's "in space")

## T17: Asteroids

It must have been decided at some point to clean up the *x3\_universe.xml* file so debris was moved into a separate file – *x3\_universe\_debris.xml*. All well and good, except some asteroids were moved into *x3\_universe\_debris.xml* as well and some debris has been added back into *x3\_universe.xml*. So now you have asteroids and debris defined in two separate files.

Take Queens Space for instance, there are 2 asteroids in this sector – the data for one resides in *x3\_universe.xml* and the other in *x3\_universe\_debris.xml*.

And to make matters worse – the occasional asteroid is defined in both – particularly in Savage Spur. I can only assume the game knows how to deal with these conflicts.

Apart from the fact that asteroids in the debris file have no *id* attribute, it seems to make no difference; they are identical to asteroids in the main universe file and can have mines places on them.

There are 11 possible attributes here:

- id** is the unique ID for the asteroid – this doesn't exist on asteroids in *x3\_universe\_debris.xml*
- t** is the type, this is always 17
- s** is the sub-type which refers to the index in *TAsteroids.pck*, whole asteroids have an *s* value between 0 and 4 – when they're blown up they become a different, smaller sub-type of asteroid right down to S10; collectable.
- x** is the 'x' position of the asteroid
- y** is the 'y' position of the asteroid
- z** is the 'z' position of the asteroid

**atype** refers to what the asteroid yields:  
0: Ore  
1: Silicon  
2: Nividium  
3: Ice

**aamount** is the actual yield quantity

**a**

**b** are rotational factors – see appendix 1

**g**

## T18: Gates

Gates are well, gates – they come in 2 flavours, warp gates and trans-orbital accelerators – ok, technically 3 since the xenon hub gates are actually a type all to themselves... oki 4 if you include destroyed gates, sheesh.

There are 13 possible attributes here:

**id** is the unique ID for the gate – this doesn't exist on gates in `x3_universe_debris.xml`

**t** is the type, this is always 18

**s** is the sub-type which refers to the index in `TGates.pck`

0: Warp Gate – North

1: Warp Gate – South

2: Warp Gate – West

3: Warp Gate – East

4: Warp Gate – destroyed

5: Trans-orbital Accelerator – North

6: Trans-orbital Accelerator – South

7: Trans-orbital Accelerator – West

8: Trans-orbital Accelerator – East

9-20: inactive and active Xenon Hub gates; you don't have to worry about these.

**x** is the 'x' position of the gate

**y** is the 'y' position of the gate

**z** is the 'z' position of the gate

**gid** is the grid id it denotes the gate direction

0: North

1: South

2: West

3: East

-1: Destroyed, this gate isn't connected to any other

**gxid** gxid and gyid denote the target sector for the gate, therefore if a gate has **gxid="1"** and **gyid="3"** it targets **Argon Prime**; obviously the gxid is the x position of the target sector and gyid is the y position.

**gyid**

**gtid** is the grid target id and specifies which facing the target gate has, therefore, normally a gate with gid 0 (north) will connect to a south gate; this target grid id is stored here.

0: North  
1: South  
2: West  
3: East  
-1: again applies to destroyed gates as they don't connect to anything

**a**

**b** are rotational factors – see appendix 1

**g**

## **T19: ?**

This *seems* to be unused.

## **T20: Specials**

Specials are, as the name implies, just special objects that float around in space; things like advertising buoys or station debris.

There are 11 possible attributes here:

**id** is the unique ID for object

**f** it's that special flag again; I don't really know what it does but there are 32 special signs with f="1" and, as before, I think it denotes that a script is running

**t** is the type, this is always 20

**s** is the sub-type which refers to the index in *TSpecials.pck*

**x** is the 'x' position of the object

**y** is the 'y' position of the object

**z** is the 'z' position of the object

**a**

**b** are rotational factors – see appendix 1

**g**

**v** this denotes the video stream to be played on advertising buoys – I think it references the first column in *Videos.txt*. A value of -1 probably loads in the default random stream from *VideoLists.txt* and a value of 0 denotes "no video" as it's used on station debris and other non-video displaying specials.



## T21: Nebulae?

I've been informed (thanks to **exogenesis** on the X3 forums) that type 21 denotes nebulae – presumably the now defunct sensor obscuring, hull eating and so on nebulae that are mentioned in the manual (and existed in X2).

## T22: ?

This *seems* to be unused.

## T23: Customisable Containers

Now this is a tricky bugger... it's a custom container object and can hold seemingly anything and how the “anything” it contains is dealt with depends on the sub-type value (**s**). Additionally, it seems possible to add this container to practically anything – including, in some circumstances, other T23 objects.

A customisable container object is actually really simple in itself and only has 2 attributes:

<b>t</b>	is the type and is always 23
<b>s</b>	is the sub-type
	1: installed shields (contains T9s)
	2: installed lasers (contains T8s)
	3: installed missiles (contains T10s)
	4: installed tech (contains T16s)
	5: docked ships (contains T7s)
	6: primary resources (any ware)
	7: products (any ware)
	8: secondary/intermediate resources (any ware) – this can be secondary resources or products that aren't “made” - for instance ship upgrades sold at trading docks.
	9: “escort ships” (in X2, thanks again to <b>exogenesis</b> ) – I tried fudging this back in by adding it to a station in my test_universe.xml file and assigning a ship to it, but it doesn't appear and just gets deleted when you export the universe back out from the Galaxy editor so I'm assuming it's completely defunct now.
	10: denotes that this is a script container – see types 24 and 26

## T24: Script Commands

I'm not even going to begin to try and fathom out all the possibilities here – needless to say that if you export a universe map **with** the script commands then anything that's running a script will have a T24 script command associated with it (wrapped inside a T23 container). The sub-type will denote the type of script that's running.

There are only 2 possible attributes here:

<b>t</b>	is the type and is always 24
<b>s</b>	is the sub-type and denotes which script is running

## T26: Script Variables

A little out of step here, jumping to 26 but that's because it fits better here.

A T26 object is a script variable, they will appear directly below a T24 script command (obviously in the same T23 container).

There are only 3 attributes for script variables:

- t** is the type and is always 26
- s** is the sub-type and denotes the variable type
- v** is the variable value

So, for instance, in Kingdoms End there is an OWP with 3 docked fighters – if we look at the Boron M4:

```
<o id="8289" t="7" s="SS_SH_B_M4" r="2">
  <o t="23" s="10">
    <o t="24" s="4"/>
    <o t="26" s="6" v="8287"/>
    <o t="26" s="4" v="0"/>
    <o t="26" s="0" v="0"/>
    <o t="26" s="0" v="0"/>
  </o>
</o>
```

You'll see that T23 is a script container (s="10") and that it has a T24 script command with the sub-type of 4 – I think this is the command "protect...".

Now the first script variable, with sub-type 6, denotes the target of the script. 8287 is the id number for the OWP that the fighter is docked at. I'm just guessing here, but I think sub-type 4 (let's call it S4) is navigational and that having a value of 0 denotes that it's docked. I only think that because I exported a ship running a "move to position" command from the galaxy editor and it had three S4 commands, presumably one for each axis, followed by an S8 which I'm assuming refers to the speed.

To be honest though – you're never going to try and hack this stuff in the xml file, it's better by far to use the galaxy editor but I've just included it as a point of interest.

## T25: Cockpits?

Slightly out of sequence here but it made more sense to keep the scripting types together.

Apparently T25 denotes cockpits (again thanks to **exogenesis**) I suspect this may be deprecated since they went the way of the dodo after X2.

## T27: ?

This *seems* to be unused.

## T28: Debris

Debris seems to have been rationalised – instead of having 4 different types all effectively denoting space junk (namely T28, T29, T30 and T31) there is now just the 1, T28, debris – the other 3 debris types have now been deprecated and the object data moved into TSpecials.pck; they are now just sub-types of T20 objects rather than having individual types of their own.

Debris can appear in either *x3\_universe.xml* or *x3\_universe\_debris.xml* and I assume the game knows how to deal with replication in these files (since it does occur with asteroids at least). I suspect the debris file was (sensibly) created to tidy up the universe file since it covers all those little lumps of rock and micrometeorites floating around – there are, in fact, 145 debris objects for Seizewell alone!

Debris basically covers asteroids that can't house mines. As such it has the same attributes:

- id** is the unique ID for the debris – this doesn't exist on debris in *x3\_universe\_debris.xml*
- t** is the type, this is always 28

**s** is the sub-type which refers to the index in *TDebris.pck*, I think.

**x** is the 'x' position of the debris

**y** is the 'y' position of the debris

**z** is the 'z' position of the debris

**atype** refers to what the debris yields when mined with a mobile drilling system:  
0: Ore  
1: Silicon  
2: Nividium  
3: Ice

**aamount** is the actual yield quantity

**a**

**b** are rotational factors – see appendix 1

**g**

### **Deprecated Debris Types**

These are old debris types that have now become T20 (Special Object) sub-types.

**T29: Dock Wrecks**

**T30: Factory Wrecks**

**T31: Ship Wrecks**

## Appendix 1: Measurements

### Distance and size

All units of size or distance such as **x**, **y** and **z** positions are to a scale of **2 millimetres**. Therefore to convert these units to kilometres – such as are displayed on the galaxy map – we need to divide these figures by 500000 and round off; to 1 decimal place for sizes over a kilometre, to 3 decimal places and multiply by 1000 (to convert to metres), on measurements under 1km.

For instance, an x, y, z conversion for an objects position might look like:

<b>x</b>	12656250	12656250/500000	25.3km
<b>y</b>	281250	281250/500000	503m
<b>z</b>	5062500	5062500/500000	10.1km

### Angles of rotation

Angles of rotation seem to be measured in some kind of scientific units – doubtless involving  $\pi$  somewhere along the way... it suffices to say that **65536** of these units make up a complete circle. This means that the possible (logical) range of numbers for **a**, **b** and **g** (the rotational angles) is +/- 0 – 65535 (depending on the direction of rotation).

The a, b and g attributes denote which plane you are rotating on. If you manage to plonk your ship facing the planet, parallel to the ecliptic plane then the **a** rotation will turn you away from the planet horizontally the **b** rotation will cause you to roll (you'll still be facing the planet) and the **g** rotation will cause you to turn away from the planet vertically.

So for a South Gate, you should something like the following:

<b>a (x/z)</b>	32768	65536/2	rotated through 180° around the y axis
<b>b (x/y)</b>	0		no rotation around the z axis
<b>g (z/y)</b>	0		no rotation around the x axis

It should be noted that a rotation of 180° around the x axis, for a gate for instance, may well look the same as the same rotation around the y axis but your gate will end up upside down!

## Appendix 2: Races

The difference races are as follows:

- 1 => argon
- 2 => boron
- 3 => split
- 4 => paranid
- 5 => teladi
- 6 => xenon
- 7 => kha'ak
- 8 => pirates
- 9 => goner
- 10 => ufo [deprecated]
- 11 => hostile race [deprecated]
- 12 => neutral race (ships that can be claimed)
- 13 => friendly race [deprecated]
- 14 => unknown (as in "unknown sector")
- 15 => n/a
- 16 => n/a
- 17 => Terran (ATF)
- 18 => Terran (USC)
- 19 => Yaki
- 20 => n/a
- ...

n/a may be reserved or more likely deprecated - these values, such as r="10" don't actually appear anywhere in either the *x3\_universe.xml* or *TShips.txt* file.

## Appendix 3: Colours

The colour is a binary switch it seems (thanks to **kryptyk** on the X2 forum).

All Blue = 255 (R=0 G=0 B=255)  
All Green = 65280 (R=0 G=255 B=0)  
All Red = 16711680 (R=255 G=0 B=0)  
White = 16777215 (R=255 G=255 B=255)

In reality this means that there is a single value for every RGB combination possible with black as 0 and white as 16777215, or  $(256^3)-1$  since it works from a zero index. This means it's all but impossible to work out any subtle tones exactly.

This should give some indication as to how fiddly binary colours are – and I'm only using steps of 50% colour (base 3) as opposed to base 256. The stepping sequence looks like:

R	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2		
G	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2
B	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2

Of course, Egosoft may have ordered their list differently or just used a simple indexed array – without some serious experimentation there's no real way to tell.

## Appendix 4: Essential Hacking Tools

If you really want to get into the nitty gritty of dismantling the X Universe and rebuilding it in your own image, you'll need the right kit. Here are the 4 most essential tools you'll need.

### The Galaxy Editor

Without the Galaxy Editor many things are much harder – instead of merely fine-tuning your galaxy in the XML files, you'd have to do everything! It is much faster and easier to use the Galaxy Editor; I don't want to even think about setting up script commands in the XML – it would be hellish.

Thankfully the Galaxy Editor is built into the game, you just need to get it up and running.

First off, if it doesn't exist already, create a folder called **maps** in the root of your X3TC install – without it you'll not be able to export any of the maps you create.

Now boot up X3TC, start or load a game and rename your pilot to **Makeanapplepiefromscratch** with the capitol M... and watch those double p's – for some reason the type input interface in X3 can be a little sticky when it comes to entering the same character several times.

If all is well, your character name won't have changed and you may have heard a little “beep” to let you know the command has been accepted.

Now if you quit out to the main screen and “Start a new game” you should see, right at the bottom of the list of possible starting scenarios, the **Galaxy Editor** option!

I'm not even going to begin to try and explain it as that would be a whole other essay all to itself – play about with it, have fun; it is actually fairly self-explanatory and probably has an easier learning curve than the game itself.

### Doubleshadow's X3 Editor and Mod Manager

These two widgets are essential for ripping the guts out of the game's pack files – without them you can't “peek under the hood” so to speak; you can't take a look at the existing XML or text files that basically define the initial state and rules for the galaxy.

To put it bluntly – you need these tools or you're stuffed ;)

Much kudos and many thanks to Doubleshadow - and, by proxy everyone he credits in developing these tools. You can grab the X3 Editor and Mod Manager from Doubleshadow's website at:

<http://www.doubleshadow.wz.cz/>

### A good text editor

You want something with lots of extra functions and ideally XML syntax highlighting and case folding.

I use Notepad++ which is a brilliant Open Source text editor able to handle pretty large files without dying. It has syntax highlighting and case folding and a very good find/replace function that allows you to use Regular Expressions (albeit fairly basic ones).

One of the biggest plus points though is that it can “bookmark” all of your search matches and then copy those bookmarks to the clipboard so that you can paste them into another file. Say for instance you want to rip all the asteroids out of the universe and debris files and paste them into a new file so you can see what's what. You just do a search on `t="17"` (with a leading whitespace), match all to place the bookmarks and then copy those bookmarks into another blank file... bingo!

Personally, for ease of searching/sorting that data I then convert the XML lines into CSV and dump it into OpenOffice Calc (spreadsheet program) – jobsagoodun.

You can download Notepad++ (for free) from:

<http://notepad-plus.sourceforge.net>

## ***Final Kudos***

Thanks to the following people who, whether with or without their knowledge, have helped me to understand what the hell is going on under the hood of the X3 galaxy.

**Doubleshadow** (obviously)

**exogenesis** – on the Egosoft X forums

**AalaarDB** – on the Egosoft X forums

**kryptyk** – on the Egosoft X forums

**DeadlyDa** – on the Egosoft X forums

And anyone else I've temporarily forgotten about – sorry :)

*Thanks all,*

Chealec (Commander, Cobra MkIII – old skool)