The EV Nova
Resource Bible

Last Revision by mcb
March 19th, 2004

> *WARNING: This information is not guaranteed to be 100% accurate. Use at your own risk. Has been known to cause cancer in laboratory animals. Caveat lector.*

by Matt Burch

Reformatted by ReinierK
August 9th, 2007

Index

Part I - Game Constants

Max Ships In System	64
Max Stellar Objects	2048
Max Systems	2048
Max Ship Classes	768
Max Stellar Classes	256
Jump Distance	1000 pixels
Max Weapon Types	256
Max Outfit Item Types	512
Max Beams On Screen	64
Max Dude Types	512
Max Ships Per Dude	16
Max Govts	256
Max Explosions On Screen	32
Max Explosion Types	64
Max Missions	1000
Num Mission Bits	10000
Max Cargo Types	256
Max Person Types	1024
Max Shots On Screen	128
Max Asteroids	16
Max Asteroid Types	16
Max Nebulae	32
Max Images Per Nebula	7
Max Simultaneous Missions	16
Max Disasters	256
Max Fleets	256
Max Ranks	128
Max Junk Types	128

Part II - Resource Descriptions

Note: Nova's resources all start at ID number 128, but the internal storage for all data file info is zero-based. Therefore, when a field in the Nova data file is said to refer to a government, stellar object, etc., it refers to it by its index number (starts at 0) unless it is specifically stated that it is referring to the ID number, which starts at 128.

Note: Some of Nova's fields refer to other resource IDs or index numbers, but their values are offset by a certain amount to indicate type. For example, the mïsn resource's AvailStel field refers to the index number of a gövt resource when its value is between 10000 and 10255. In cases like this, it is necessary to add to or subtract from the field in order to force the value into the proper range: in this instance, you'd subtract 10000 to find the index number of the gövt.

Note: Any resources in an Nova plug-in file automatically replace same-numbered resources in Nova's main files. Resources are loaded from the "Nova Files" folder first and then resources from the "Nova Plug-Ins" folder are loaded.

A quick word about control bits and scripting in EV Nova:

Mission bits exist as they did in previous incarnations of EV. This time around, there are 10,000 bits available for your use, and they are accessed much differently from previous versions. For this reason, they are referred to as Nova control bits (ncb's). Control bits are accessed through logical expressions that allow much more powerful and logical mission scripting. These expressions are divided into two types - test and set:

`Test expressions`:

These are boolean expressions that are used to determine when something happens; for example, when a mission is to be offered, or when a particular ship should be made available for purchase. In general, if the logical expression defined in a given test expression field evaluates to be true (nonzero), the associated property will be activated (mission becomes available, ship appears, etc.).

The following terms and **`operators`** are supported: (capitalization doesn't matter)

`Bxxx`	Lookup the value of control bit xxx. Bits are numbered from b0 to b9999.
`Pxxx`	Check if the game is registered ([P]aid for) ... evaluates to 1 if the game is registered or is unregistered but less than xxx days have elapsed. Evaluates to 0 only if unregistered for more than xxx days.
`G`	Lookup the player's gender - 1 if male, 0 if female
`Oxxx`	Returns 1 if the player has at least one of outfit item ID xxx, 0 if not
`Exxx`	Returns 1 if the player has explored system ID xxx, 0 if not
|	Logical or operator
&	Logical and operator
`!`	Logical negation operator
`( )`	Parenthetical enclosure

Some examples:
```
b13 & (b15 | !b72)
!(B42 | B53) & b103
```

Note that since the Nova evaluator is fairly primitive, it may do unpredictable things if you give it an expression like `b1 & b2 | b3` ... instead, use proper parentheses to make it `b1 & (b2 | b3)` or `(b1 & b2) | b3`, as appropriate.

Also note that if you leave the field for a test expression blank, it will evaluate to true as a default.

Also note: The Oxxx operator also considers any carried fighters that are deployed when it examines the player's current list of outfits, although this feature may be confused if presented with a universe that includes multiple fighter bay weapons that launch the same ship type, or different outfits that grant the same fighter bay ammo.

Set expressions:

These are simpler than the test expressions... basically all you are doing here is listing what bits you want to be modified when the expression in a given field is invoked. This will happen when the player does something (completes a mission, buys an item, etc.) as defined by the other resources. The syntax of set expressions is best illustrated by an example:

```
b1 b2 !b3 ^b4
```

In this set expression, bits 1 and 2 will be set, bit 3 will be cleared, and bit 4 will be toggled to the opposite of whatever it was previously. No parentheses are supported for set expressions. Note that if you leave a set expression blank, no control bits will be altered.

One other feature of the set expression is the ability to make random decisions. By specifying R(<op1> <op2>)   you can make Nova randomly pick one of the two possible choices and execute it, skipping the other one. For example:

```
b1 R(b2 !b3)
```

...this expression will set bit 1, and then *either* set bit 2 *or* clear bit 3, but not both at once. Which operation will be picked is completely random, which allows for the design of interesting mission strings that branch unpredictably.

There are also a number of **other operators** that allow you to do many interesting things:

Axxx	if mission ID xxx is currently active, abort it.
Fxxx	if mission ID xxx is currently active, cause it to fail.
Sxxx	start mission ID xxx automatically.
Gxxx	grant one of outfit item ID xxx to the player
Dxxx	remove (Delete) one of outfit item ID xxx from the player
Mxxx	move the player to system xxx. The player will be put on top of the first stellar in the system, or in the centre of the system if no stellars exist there.
Nxxx	move the player to system xxx. The player will remain at the same x/y coordinates, relative to the centre of the system.
Cxxx	change the player's ship to ship type (ID) xxx. The player will keep all of his previous outfit items and won't be given any of the default weapons or items that come with ship type xxx.
Exxx	change the player's ship to ship type (ID) xxx. The player will keep all of his previous outfit items and will also be given all of the default weapons and items that come with ship type xxx.
Hxxx	change the player's ship to ship type (ID) xxx. The player will lose any nonpersistent outfit items he previously had, but will be given all of the default weapons and items that come with ship type xxx.
Kxxx	activate rank ID xxx.
Lxxx	deactivate rank ID xxx.
Pxxx	play sound with ID xxx.
Yxxx	destroy stellar ID xxx.
Uxxx	regenerate (Un-destroy) stellar ID xxx.

Qxxx	make the player immediately leave (absquatulate) whatever stellar he's landed on and return to space, and show a message at the bottom of the screen. The message is randomly selected from the STR# resource with ID xxx, and is parsed for mission text tags (e.g. <PSN> and <PRK>) but not text-selection tags like those above (e.g. {G "he" "she"}) (see dësc and mïsn resource descriptions for more examples)
Txxx	change the name (Title) of the player's ship to a string randomly selected from STR# resource ID xxx. The previous ship name will be substituted for any '*' characters which are encountered in the new string.
Xxxx	make system ID xxx be explored.

The spïn resource

Spin resources contain sprite info for simple graphical objects. Whenever Nova needs to load a set of sprites for a particular object, it looks at that object's spin resource, which in turn tells the game how to load the object's sprites. Nova sprites are stored as paired sprite and mask PICT resources, or as rleD/rle8 resources. The sprites in each PICT are arranged in a grid, which can be of any size. The spin resource tells Nova what shape and size the sprites' grid is. Spin resources have the following fields:

SpritesID	ID number of the sprites' PICT resource (or the ID of the rleD/rle8 resource).
MasksID	ID number of the masks' PICT resource.
xSize	Horizontal size of each sprite.
ySize	Vertical size of each sprite.
xTiles	Horizontal grid dimension .
yTiles	Vertical grid dimension .

Spin resources have certain reserved **ID numbers**, which correspond to different types of objects:

400-463	Explosions.
500	Cargo boxes.
501-504	Mini-asteroids for mining.
600-605	Main menu buttons.
606	Main screen logo.
607	Main screen rollover images.
608-610	Main screen sliding buttons.
650	Target cursor.
700	Starfield.
800-815	Asteroids.
1000-1255	Stellar objects.
3000-3255	Weapons.

It is important to note that the ID numbers of the PICT/rleD/rle8 resources are non-critical, as Nova looks at the spin resources to find the sprites, and not at the actual PICT/rleD/rle8 ID numbers themselves.

The shän resource

Shan (ship animation) resources contain sprite info for ship graphics, which are too complex for the more rudimentary spin resource.

BaseImageID	The resource ID of the basic sprite images for this ship.
BaseMaskID	The ID of the corresponding sprite masks (ignored if the base image is an rleD/rle8 resource).
BaseSetCount	The number of sprite sets for the basic sprite images. A sprite set is usually 36 sprite images, and the graphics for all of a ship's basic sprite sets are stored in the same PICT/rleD/rle8 resource, referred to in BaseImageID.
BaseXSize	The X size of each basic sprite image.
BaseYSize	The Y size of each basic sprite image.
BaseTransp	The inherent transparency of the basic sprite images, from 0 (no transparency) to 32 (fully transparent).
AltImageID	The resource ID of the alternating sprite images for this ship. Sprites from the alt sprite sets can be displayed on top of the basic sprite for the ship, cycling through each available sprite set at a rate defined in the Delay field, below. Set to zero if unused.
AltMaskID	The corresponding mask ID. Set to zero if unused.
AltSetCount	The number of sprite sets for the alternating sprites.
AltXSize	The X size of the alt sprite image.
AltYSize	The Y size of the alt sprite image.
GlowImageID **GlowMaskID** **GlowXSize** **GlowYSize**	Engine glow.
LightImageID **LightMaskID** **LightXSize** **LightYSize**	Running lights.
WeapImageID **WeapMaskID** **WeapXSize** **WeapYSize**	Weapon effects.
ShieldImageID **ShieldMaskID** **ShieldXSize** **ShieldYSize**	Shield bubble (shield sprite have a number of frames exactly equal to 1, FramesPer, or BaseSetCount*FramesPer).

Flags

0x0001	Extra frames in base image are used to display banking. The first set of sprites is used for level flight, the second for banking left, and the third for banking right.
0x0002	Extra frames in base image are used for animated ship parts such as for folding/unfolding wings. The sprites will be cycled upon landing, taking off, and entering/exiting hyperspace.
0x0004	The second set of frames in the base image are displayed when the ship is not carrying any of its KeyCarried type ships onboard.
0x0008	Extra frames in base image are shown in sequence, just like the sprites in the alternating image. The AnimDelay field has the same effect in this case.
0x0010	Stop the ships' animations when it is disabled.
0x0020	Hide alt sprites when the ship is disabled.
0x0040	Hide running light sprites when the ship is disabled.
0x0080	Ship unfolds when firing weapons, and folds back up when not firing.
0x0100	Adjust ship's visual presentation to correct for the skew caused by graphics that are rendered highly off-axis from vertical. This uses the ship's UpCompressY and DnCompressY fields to interpolate the proper sprite frame to display based on the ship's actual heading. Use this with caution, as it tends to cause very jerky ship rotation and is mostly included as a curiosity.

Note that the first four flags in this field are mutually exclusive - i.e. you can have a ship that banks, unfolds, changes appearance when it is carrying a certain other ship type, or animates in sequence, but these effects can't be combined. The only exception is that having both flags 0x0001 and 0x0002 set is treated specially - it results in a ship whose extra frames are used for banking and which always displays its engine glow when it is turning, whether or not it is actually accelerating. (This something that got thrown in at some point when I realized that it would be necessary to have in order to replicate the behaviour of a certain type of ship from a certain TV show).

AnimDelay The delay between frames of the sprite animations, in 30ths of a second.

WeapDecay The rate at which the weapon glow sprite fades out to transparency, if applicable. 50 is a good median number - lower numbers yield slower decays.

FramesPer The number of frames for one rotation of this ship - usually 36 is a good number, but larger ships can benefit from having more frames per rotation to make their turning animation look smoother. Be sure this value is equal to the actual number of frames per revolution in your images, or bad things will happen!

BlinkMode

0 or -1	Ignored.
1	Square-wave blinking: BlinkValA is the light on-time. BlinkValB is the delay between blinks. BlinkValC is the number of blinks in a group. BlinkValD is the delay between groups.
2	Triangle-wave pulsing: BlinkValA is the minimum intensity (1-32). BlinkValB is the intensity increase per frame, x100. BlinkValC is the maximum intensity (1-32). BlinkValD is the intensity decrease per frame, x100.
3	Random pulsing: BlinkValA is the minimum intensity (1-32). BlinkValB is the maximum intensity (1-32). BlinkValC is the delay between intensity changes. BlinkValD is ignored.

GunPosX	Here you can set the exit points on the ship sprite for four.
GunPosY	different classes of weapons. Note that The "Gun" "Beam" etc.
TurretPosX	designations are for convenience only, since which set of
TurretPosY	weapon exit points is used by a given weapon are defined in
GuidedPosX	that weapon's ExitType field. The x & y positions of each
GuidedPosY	weapon exit point are measured in pixels from the centre of the
BeamPosX	ship when the ship is pointing straight up (frame index 0). See
BeamPosY	the next four fields if you need to account for any perspective corrections in your sprites.
UpCompressX	If you have ship sprites that are rendered at an angle, these
UpCompressY	fields are used to correct for the ships perspective when
DnCompressX	calculating the weapon exit points (above). If the ship is
DnCompressY	pointing generally "up" (heading is 0-90 or 270-359) then UpCompressX/Y are used; if the ship is pointing generally "down" (heading is 91-269 degrees) then DnCompressX/Y are used. These values are divided by 100 and then multiplied by the rotated x & y values in the weapon exit point fields to apply a rough correction factor, so values less that 100 will bring the exit points in closer to the ship and values greater than 100 will move the exit points farther out. Experimentation is the best way to learn how this works. Values of zero are interpreted the same as a value of 100, so you can leave this field set to zero if unused.
GunPosZ	Here you can set further weapon exit point offsets in order to
TurretPosZ	compensate for skew caused by the z position of a ship
GuidedPosZ	graphic's weapon exit point. These values are added to a
BeamPosZ	shot or beam's position after the weapon exit point x & y offsets and the x & y compression factors have been applied, so the effect of these values is not scaled. Positive values here move up the screen, negative values move down the screen. (this is a lot easier to use with the editor than it is to describe).

Note: if you use engine glows or running lights, you must have the same number of engine glow and/or running light frames as base frames (including banking frames!) or Nova will choke.

The böõm resource

The boom resource is used to customize the various explosion types. Nova supports up to 64 different explosion types. The graphics for the explosions are loaded from spïn resources 400-463, the sounds are loaded from snd resources 300-363, and the behaviour of each explosion type is contained within böõm resources 128-191. Each böõm resource contains three fields:

`FrameAdvance`	The rate at which the explosion will animate - a value of 100 will cause each frame of the explosion to appear for exactly one frame of the game animation, and lower values will stretch out the explosion animation and make it stay onscreen longer.
`SoundIndex`	The index (0-63) of the explosion sound to associate with this explosion type, or -1 for a silent explosion. Usually you'd set this to either -1 or to the same value as the explosion index itself (e.g. 1 for böõm resource 129, etc.) but if you want to use the same sound for two different explosion types, you can do that with this field.
`GraphicIndex`	The index (0-63) of the explosion graphic to associate with this explosion type. Usually you'd set this to the same value as the explosion index itself (e.g. 1 for böõm resource 129, etc.) but if you want to use the same graphic for two different explosion types (like the small weapon explosion and the ship-breaking-up explosion) you can do that with this field.

The chär resource

The chär resource is used to allow multiple entry points into the scenario's storyline, by letting the player pick a "character template" from a list when a new pilot is started. The player is presented with a list of all available chär resources when a new pilot is created, and must choose one (but only one). This allows different character types to have different ships, legal records, etc. at the start of the game, and also allows for the setting-up of mission stuff by way of a control bit set string that is evaluated when a new pilot is started.

Cash	The amount of money a player gets when starting out with this character type.
ShipType	ID number of the starting ship type.
System1-4	ID numbers of up to four possible starting systems for the player. The player will randomly be placed in one of these systems when starting out. Set to -1 if unused (if all four of these fields are set to -1, the player will be placed in system ID 128 as a default).
Govt1-4 & **Status**1-4	For each of the governments whose ID is entered in a Govt1-4 field, the player's legal status in systems owned by that government or one of its allies is set to the value in the corresponding Status field. For systems owned by an enemy of the govt identified in the Govt field, the player's legal status is set to the negative value of the number in the corresponding Status field. Set unused Govt fields to -1.
Kills	The player's starting combat rating.
IntroPict1-4	IDs of up to four PICT resources to show in sequence when the player starts out with a new pilot of this type. Set to -1 if unused.
PictDelay1-4	Maximum delay time to display each of the four above pictures, in seconds.
IntroTextID	The ID of the dësc resource to show when the player starts out with a new pilot of this type. Along with the IntroPictID field, this allow you to have different opening sequences for each pilot type (useful to show different sides of the same issue, for example). Using the dësc resource's MovieFile field here lets you have an introduction movie. The intro text (and any associated movie) is displayed after the above PICT resources are shown.
OnStart	A control bit set string that is called when the player starts out with a new pilot of this type.
Flags	
0x0001	Denotes the "default" chär resource, which will be automatically selected in the popup menu. If more than one chär resource has this bit set (there shouldn't be) the one with the lowest ID will be considered the default.
StartYear	The starting year of the game.
DatePrefix	String that is appended to the start of the date whenever it's displayed.
DateSuffix	String that is appended to the end of the date whenever it's displayed.

The cölr resource

The cölr resource allows you to customize some game-wide interface options.

ButtonUp	Normal button text colour.
ButtonDown	Pressed button text colour.
ButtonGrey	Greyed-out button text colour.
MenuFont	Main menu font name.
MenuFontSize	Size of main menu font.
MenuColor1 & 2	Bright & dim colours for main menu.
GridDim	Shipyard/outfit dialog grid colour.
GridBright	Shipyard/outfit dialog selection square colour.
ProgressBar	Position and shape of the loading progress bar, relative to the centre of the window.
ProgBright	Bright progress bar colour.
ProgDim	Dim progress bar colour.
ProgOutline	Progress bar outline colour.
Button1x & y Through Button6x & y	Position of the six main menu buttons, relative to the top left corner of a 1024x768 main menu background.
FloatingMap	Floating hyperspace map / escort menu border colour.
ListText	List text colour.
ListBkgnd	List background colour.
ListHilite	List hilite colour.
EscortHilite	Escort menu item hilite colour.
ButtonFont	Button font name.
ButtonFontSz	Size of button font.
LogoX & Y	Logo animation x/y position.
RolloverX & Y	Rollover animation x/y position.
Slide1x & y Through Slide3x & y	Sliding button x/y positions.

The various interface buttons that appear are drawn on the fly. Nova uses PICT resources 7500-7502 for the left, centre, and right pieces of the "up" buttons, PICT resources 7503-7505 for the "down" button pieces, and PICT resources 7506-7508 for the greyed-out button pieces. Corresponding mask images are stored in PICTs 7600-7608. STR# resource 150 is used to store the text that appears on each button type.

Note that all colour fields in the cölr resource are encoded the same as HTML colours, and that only the first cölr resource is loaded.

VI The crön resource

Cron resources are used to define time-dependent events that occur in a manner that is invisible to the player but can cause interesting things to happen in the universe, via the manipulation of control bits. With it, you can create such things as:

- an event that occurs periodically during the course of the game;
- an event that occurs at some fixed date during the game, as part of the story's set script;
- an event, triggered by the actions of the player, that occurs after some fixed or random interval;
- etc.

FirstDay	The first day of the month (1-31) on which the cron event can be activated. If you set this to 0 or -1, this field will be ignored and only FirstMonth and FirstYear will be considered.
FirstMonth	The first month of the year (1-12) on which the cron event can be activated. Set to 0 or -1 for this to be ignored.
FirstYear	The first year in which the cron event can be activated. Set to 0 or -1 for this to be ignored.
LastDay	The last day of the month (1-31) on which the cron event can be activated. Set to 0 or -1 for this to be ignored.
LastMonth	The last month of the year (1-12) on which the cron event can be activated. Set to 0 or -1 for this to be ignored.
LastYear	The last year in which the cron event can be activated. Set to 0 or -1 for this to be ignored.
Random	The percent chance that the cron event will be activated during the date range defined above. Set to 100 for the event to be activated as soon as it can be.
Duration	The duration during which the event is active, in days. If this is set to zero, the event will start and end on the same day, i.e. its OnStart and OnEnd scripts will be run at the same time.
PreHoldoff	The number of days to "hold" the event in a waiting state after it is activated and before it starts. Set this to zero to have the event start immediately when it is activated.
PostHoldoff	The number of days to hold the event in a waiting state after it ends and before it is deactivated. This is used to keep a repeating event from being activated immediately after it has just happened. Set this to zero to have the event be deactivated immediately after it ends.
Flags	
0x0001	Continuous, iterative cron entry - keep evaluating the cron's OnStart field until the EnableOn expression is no longer true or the constraints of the Require fields are no longer met. This can create infinite loops, so be careful!
0x0002	Continuous, iterative cron exit - keep evaluating the cron's OnExit field until the EnableOn expression is no longer true or the constraints of the Require fields are no longer met. This can create infinite loops, so be careful!
EnableOn	A control bit test string that is used to determine whether the cron event is eligible to be activated or not. Leave this blank if you are creating an event whose activation doesn't depend on the state of any control bits.

`OnStart`	A control bit set string that is called when the cron event starts, after waiting through the PreHoldoff time, if any.
`OnEnd`	A control bit set string that is called when the cron event ends.
`Contribute` `Contribute`	When the cron event is active, these two Contribute fields together form a 64-bit flag that is subsequently combined with the Contribute fields from the player's ship and the other outfit items in the player's possession, to be used with the Require fields in the outf and misn resources.
`Require` `Require`	These two Require fields together form a 64-bit flag that is logically and'ed with the Contribute fields from the player's current ship and outfit items. Unless for each 1 bit in the Require fields there is a matching 1 bit in one or more of the Contribute fields, the cron will not be activated. Leave these set to zero if unused.

`NewsGovt``1-4`
`GovtNewsStr``1-4`
	On planets or stations that are allied with the government whose ID is given by one of the NewsGovt fields, a string will be randomly selected from the STR# resource whose ID is given by the corresponding GovtNewsStr field, and will be displayed as news while the cron event is active. This allows you to let up to four different governments (and their allies) have their own "local news" for a given cron event. Set unused NewsGovt and GovtNewsStr fields to -1.

`IndNewsStr`	The ID of a STR# resource from which to randomly select a string to be displayed in the news dialog while this cron event is in progress, if it doesn't have any applicable local news. Set to -1 for no independent news.

Some notes:

1. Setting any of the above date fields to 0 or -1 effectively makes that field a wildcard field, which will match to anything.

2. If you want an event with a wide possible date range to be guaranteed to never run more than once, make it set a control bit in its OnEnd script that will prevent it from subsequently being eligible for activation.

3. The 'M' and 'N' control bit set string operators should probably not be used in conjunction with cron events, unless you really want to confuse the player by moving him around at seemingly random times.

4. Local news always takes precedence over independent news, even if there is no corresponding news string to display (the STR# ID must still be greater than zero to not be ignored). You can use this to make everyone in the universe except a particular government or set of governments report on something, for example.

The dësc resource

Desc resources store null-terminated text strings (descriptions) that are used by Nova in a variety of places. For some desc resources, Nova looks for a certain reserved ID number. Other desc resources are pointed to by fields in other resources, so their ID numbers are not necessarily fixed, and can be set to virtually anything by the scenario designer. The reserved desc ID numbers, along with the maximum length for each type, are below:

`128-2175`	Stellar object descriptions, shown when landed on a planet.
`3000-3511`	Outfit item descriptions, shown in ship outfitting dialog.
`4000-4999`	Mission descriptions, shown in mission dialog.
`13000-13767`	Ship class descriptions, shown in the shipyard and requisition-escort dialog.
`13999`	Message shown after the player uses an escape pod.
`14000-14767`	Ship pilot descriptions, shown in the hire-escort dialog.
`32760-32767`	Reserved.

If you wish, you can make a dësc resource mutable via control bits - embedding a special sequence of characters into the dësc resource will instruct Nova to change the contents of the text on the fly. This sequence is delimited (marked) by the characters "{" and "}", and follows this format:

```
{bXXX "string one" "string two"}
```

Where "XXX" is be replaced by the index of the control bit you wish to test. You can add in a "!" character before the "bXXX" test in order to negate the result of the test, but unlike the control bit test strings, you cannot perform compound tests in a dësc resource - i.e., no testing of multiple bits at a time.

If the bit test (after being negated, if the "!" character is present) evaluates to true, the first string will be substituted in place of all the characters between (and including) the "{" and "}" characters. If the bit test evaluates to false and there is a second string in the expression, that second string will be substituted. If there is no second string, nothing will be substituted.

For example, consider this dësc resource:

```
This is a {b001 "great and terrific" "lousy, terrible"} example.
```

...if bit 001 is set, the output will be "This is a great and terrific example." If bit 001 is not set, the output will be "This is a lousy, terrible example.".

Also note that if you want to include a quotation mark (") character in either of the two strings, use standard C syntax to do it:

```
My name is {b002 "Dave \"pipeline\" Williams"}
```

This is also works with the player's gender - for example:

```
This is a test string and the player is {G "a male character" "a female
pilot"}.
```

...in this case, the G character signifies that the following text is mutable based on the player's gender; if the player is male, the first string is used, otherwise the second string is used. Note that the "!" token works here as usual.

You can also change the text based on whether or not the player is registered:

```
This is a test string you {P "have paid" "haven't paid"}.
```

...in this case, the P character signifies that the following text is mutable based on whether or not the game is registered; if the player has registered, the first string is used, otherwise the second string is used. Note that the "!" token works here as usual, and you can also append a number to the P character to specify a number of days, just as you can with the ncb Pxxx test operator.

Besides the Description field in the dësc resource, there are some additional fields:

Graphic	This is used to include graphics in mission briefings. If you put in the ID of a valid PICT resource, Nova will display that image along with the description text when it displays a mission dialog box (with the exception of the Mission Computer and Mission Info dialogs).
MovieFile	The name of a QuickTime movie file to display before the briefing dialog appears. This file must reside either in "Nova Files" or "Nova Plugins".

Flags

0x0001	Show the movie after the briefing instead of before.
0x0002	Show movie at double-size.
0x0004	Cinematic movie - blank background and fade screen before and after .

VII The düde resource

A dude resource can be thought of as a container for ships that share certain characteristics. Any ship of a given dude class will have that dude class's AI type and governmental affiliation, and will yield the same types of booty when boarded. In a dude resource, up to 16 different ship classes can be pointed to, with a probability set for each ship class. The result of all this is that, in other parts of Nova's data file, you can point to a dude class and know that Nova will create a ship of the proper AI type and governmental alignment, and will pick the new ship's type based on the probabilities you set in the dude resource. The dude resource's fields are:

AIType	Which type of AI to use for ships of this dude class (see below). If you set this to 0, each ship will use its own inherent AI type.
Govt	The ID number of the dude class's government, or -1 for independent.
Booty	Flags that define what you'll get when you board a ship of this dude class. (see below)
InfoTypes	What kind of info to display when hailed.
0x1000	Good prices.
0x2000	Disaster info.
0x4xxx	Specific advice (the lower 12 bits of this value are added to 7500 to get the ID of the STR# resource from which to get the quote).
0x8000	Generic govt hail messages.
Flags	
0x0001	Carries food when plundered.
0x0002	Carries industrial goods.
0x0004	Carries medical supplies.
0x0008	Carries luxury goods.
0x0010	Carries metal.
0x0020	Carries equipment.
0x0040	Carries money (amount depends on the ship's purchase price).
0x0100	Ships of this dude type can't be hit by the player and their shots can't hit the player (useful for things like AuxShip mission escorts, etc.).
ShipType (x16)	These fields contain the ID numbers of up to 16 different ship classes. Set to 0 or -1 if unused.
Probability (x16)	These fields set the probability that a ship of this dude class will be of a certain ship type.

The four different AI types are:

1 - Wimpy Trader	Visits planets and runs away when attacked
2 - Brave Trader	Visits planets and fights back when attacked, but runs away when his attacker is out of range.
3 – Warship	Seeks out and attacks his enemies, or jumps out if there aren't any.
4 – Interceptor	Seeks out his enemies, or parks in orbit around a planet if he can't find any. Buzzes incoming ships to scan them for illegal cargo. Also acts as "piracy police" by attacking any ship that fires on or attempts to board another, non-enemy ship while the interceptor is watching.

You can set different combinations of booty to be had from ships of a certain dude class by ORing different bits into the dude's Booty field. If a dude class has a booty flag of 0x0000, then you can't get anything from the ship, and you're told that you were "repelled while attempting to board" it. The different booty flags are documented above

> *Note: unlike previous versions of the EV engine, Nova can handle dude resources that are not completely filled - so, while the dude resource can handle references to as many as 16 different ship types, not all fields have to be used. If you do this, remember to set the corresponding ShipType fields to 0 or -1 to indicate that they're unused.*

IX The flët resource

A flet resource defines the parameters for a fleet, which is a collection of ships that can be made to appear randomly throughout the galaxy.

LeadShipType	ID of the fleet's flagship's ship class.
EscortType (x4)	IDs of the flagships escorts' ship classes. If you don't want to use four different escort types, you should still set the unused fields to a valid ship class ID. (you can set the min & max fields to 0 and just have the extra ships not appear).
Min (x4)	The minimum number of each type of escort to put in the fleet.
Max (x4)	The maximum number of each type of escort to put in the fleet.
Govt	ID of the fleet's government, of -1 for none.
LinkSyst	Which systems the fleet can be created in.
-1	Any system.
128-2175	ID of a specific system.
10000-10255	Any system belonging to this specific government.
15000-15255	Any system belonging to an ally of this govt.
20000-20255	Any system belonging to any but this govt.
25000-25255	Any system belonging to an enemy of this govt.
AppearOn	A control bit test field that will cause a given fleet to appear only when the expression evaluates to true. If this field is left blank it will be ignored.
Quote	Show a random string from the STR# resource with this ID when the fleet enters from hyperspace. Any occurrences of the character '#' in this string will be replaced with a random digit (0-9).
Flags	
0x0001	Freighters (InherentAI <= 2) in this fleet will have random cargo when boarded.

The gövt resource

A govt resource defines the parameters for a government, which is in turn defined as "any collection of ships and planets that react collectively to the actions of the player and other ships." Governments keep track of how they feel toward you, and they can also have set enemies and allies. The govt resource's fields are:

VoiceType	Sets this government's voice type, used for when you have a ship of this government as your escort (i.e. an escort with an inherent attributes govt field that points to this govt). There can be up to eight different voice types, numbered 0-7. The voice resources are loaded from 'snd ' resources as follows:

`1000-1009`	Voice type 0 acknowledgement speech.
`1010-1019`	Voice type 0 targeting speech.
`1020-1029`	Voice type 0 victory speech.
`1100-1109`	Voice type 1 acknowledgement speech.
`1110-1119`	Voice type 1 targeting speech.
`1120-1129`	Voice type 1 victory speech.
`...etc, up to:`	
`1700-1709`	Voice type 7 acknowledgement speech.
`1710-1719`	Voice type 7 targeting speech.
`1720-1729`	Voice type 7 victory speech.

If a particular set of voices (i.e. the acknowledgement, targeting, or victory sounds for a given voice type) contains an even number of sound resources, Nova will only use either the even- or odd-numbered sounds for each particular ship. The Nova scenario uses this to implement both male and female voices for certain governments. To let each ship decide for itself whether to use even- or odd-numbered sounds, set VoiceType to between 0 and 7. To force ships to use only odd-numbered sounds, set VoiceType to between 1000 and 1007. To force ships to use only even-numbered sounds, set VoiceType to between 2000 and 2007.

(None of the preceding paragraph applies if Nova doesn't have an even number of sounds to work with)

If you don't want any ships of a government to use speech, set that govt's VoiceType to -1. Also note that ship types with no inherent attributes govt defined (see shïp section for more information) will always use voice type 0.

Flags & **Flags2**	Sets a variety of characteristics (see below).
CrimeTol	The maximum amount of evilness the player can accumulate before warships of this govt start to beat on him.
ScanFine	If the player is caught carrying an illegal (and non-mission-related) cargo or item by a ship of this govt and he isn't yet evil enough to attack (i.e. his legal status in the current system isn't below CrimeTol) then he will be fined the amount in this field.

`1 and up`	fine this amount.
`0`	no fine, just a warning.
`-1 and below`	fine this % of the player's cash (-5 is 5%, etc).

SmugPenalty	The amount of evilness a player gains for being detected smuggling illegal cargo (defined in a misn resource) past this government's ships.
DisabPenalty	The amount of evilness for disabling one of this govt's ships.

BoardPenalty	Evilness from pirating one of this govt's ships.
KillPenalty	Evilness from killing this govt's ships.
ShootPenalty	Evilness from shooting one of this govt's ships (currently ignored).
InitialRec	The player's initial legal record in systems controlled by this govt (0 is neutral, positive is good, negative is bad)
MaxOdds	The maximum combat odds ships of this govt will consider favourable. Combat odds are calculated by summing the strengths of the ship's enemies (where a ship's strength is taken from the Strength field in the shïp resource, and modified from between 30% and 100% of that value depending on the ship's present shield stat) and comparing it to the sum of the strength of the ship's friends. A value of 100 in this field represents 1-to-1 combat odds, and will cause a ship of this govt not to attack unless it calculates that it is as strong, or stronger than, its enemy. A value of 200 represents 2-to-1 combat odds, meaning that ships of this govt won't engage if they are outnumbered by more than 2-to-1. 300 means that ships of this govt won't engage a group of enemies more than 3x stronger than them and their friends, etc.
Class1-Class4	Allows you to assign this govt to up to four different "classes", which are simply arbitrary groupings of govts that you can use to flexibly assign allies and enemies. Two govts of the same class are not inherently allied unless one of them has that same class number set in one of its Ally fields. Set unused class fields to -1.
Ally1-Ally4	The number of up to four classes that this govt will be allied with. Set to -1 if unused.
Enemy1-Enemy4	The number of up to four classes that this govt will be enemies with. Set to -1 if unused.
Interface	ID of an ïntf resource to use when the player is flying a ship whose inherent attributes govt or inherent combat govt is equal to this govt type. Values less than 128 will be interpreted as 128.
NewsPic	ID of a PICT resource to use as the background of the news window when the player is on a planet or station owned by this govt. Values less than 128 are ignored and the standard independent/generic news background (ID 9000) is substituted instead.

Doing evil deeds to one government will improve your rating with its enemies, and vice versa. Allied governments also communicate your actions, so attacking one government will make its allies hate you too.

The different **bits** that can be set in a govt's Flags field are:

0x0001	Xenophobic (Warships of this govt attack everyone except their allies. Useful for making pirates and other nasties.)
0x0002	Ships of this govt will attack the player in non-allied systems if he's a criminal there (useful for making one govt care only about the player's actions on its home turf, while another is nosy and enforces its own laws everywhere it goes).
0x0004	Always attacks player.
0x0008	Player's shots won't hit ships of this govt.

0x0010	Warships of this govt will retreat when their shields drop below 25% - otherwise they fight to the death.
0x0020	Nosy ships of other non-allied governments ignore ships of this govt that are under attack.
0x0040	Never attacks player (also, player’s weapons can't hit them).
0x0080	Freighters (i.e. AiTypes 1 and 2) for this particular government have 50% of the standard InherentJam value for warships (AiType 3) of the same government.
0x0100	'pers' ships of this govt won’t use escape pod, but will act as if they did.
0x0200	Warships will take bribes.
0x0400	Can't hail ships of this govt. (if a ship type has an inherent attributes govt which includes this flag, all ships of that type will inherit this property)
0x0800	Ships of this govt start out disabled (derelicts). Note that ships of other governments don't care if you attack or board derelict govt ships.
0x1000	Warships will plunder non-mission, non-player enemies before destroying them.
0x2000	Freighters will take bribes.
0x4000	Planets of this govt will take bribes
0x8000	Ships of this govt taking bribes will demand a larger percentage of your cash supply, and their planets will always take bribes (useful for pirates).

Flags2:

0x0001	When hailing ships of this govt, the request assistance / beg for mercy button is disabled and the govt is not talkative.
0x0002	This govt is considered "minor" for the purposes of drawing the political boundaries on the map.
0x0004	This govt's systems don't affect the political boundaries on the map.
0x0008	Ships of this govt don't send distress messages and don't respond with greetings when hailed (if a ship type has an inherent attributes govt which includes this flag, all ships of that type will inherit this property)
0x0010	Roadside Assistance - Ships of this govt will always repair or refuel the player for free.
0x0020	Ships of this govt don't use hypergates.
0x0040	Ships of this govt prefer to use hypergates instead of jumping out.
0x0080	Ships of this govt prefer to use wormholes instead of jumping out.

The SkillMult field allows you to apply a global multiplier to the skill levels of ships that belong to this government. A value of 100 will cause this government's ships to be just as skilled as any other ships of the same type. A value of 50 will cause this govt's ships to only be 50% as skilled, and a value of 150 will cause this govt's ships to be 50% more skilled as the stock ship is rated. This allows you to create governments whose pilots are more highly trained than the stock pilots, so they can gain extra speed and acceleration from their ships. Values in this field of less than 1 are ignored.

ScanMask	This is a 16-bit flags field that is used in conjunction with the ScanMask field in the mïsn resource. If any of the 1 bits in a government's ScanMask field match any of the 1 bits in a mission's ScanMask field, that government will consider the mission's cargo illegal. Set to zero if unused.
Require	These two Require fields together form a 64-bit flag that is logically and'ed with the Contribute fields from the player's current ship and outfit items. If for each 1 bit in the Require fields there is not a matching 1 bit in one or more of the Contribute fields then you won't be allowed to visit any planets or stations owned by this govt - this is useful for making travel permits, for example. Leave these set to zero if unused.

InhJam1-4	The government's inherent jamming ability for each of the four jamming types, from 0 to 100%.
MediumName	The government's name in medium length, which is primarily used in the string "Sensors detect xxx reinforcement fleet approaching."
Colour	The government's theme colour, encoded the same as HTML colours. (00RRGGBB).
ShipColor	The theme colour for this government's ships, encoded the same as HTML colours (00RRGGBB). You should use this sparingly, as it gets annoying when you have dozens of multicoloured variants of some ship type flying around all at once. Set to 00000000 if unused (which is the same as 00FFFFFF, or "pure white paint" i.e. no shading).
CommName	The short string to show for ships of this government when they are hailed by the player.
TargetCode	The short string to show in the player's target display when a ship of this government is targeted.

The jünk resource

Junk resources store info on specialized commodities that can be bought and sold at a few locations. The fields are:

SoldAt1-8	ID number of the stellar object where the commodity is sold. Set to 0 or -1 if unused.
BoughtAt1-8	ID number of the stellar object where the commodity is purchased. Set to 0 or -1 if unused.
BasePrice	The average price of the commodity (works much like the base prices for "regular" commodities).
Flags	Misc flag bits.
0x0001	Tribbles flag - When in your cargo bay, the commodity multiplies like tribbles.
0x0002	Perishable - When in your cargo bay, the commodity gradually decays away.
ScanMask	If this is an "illegal" cargo type, this is used in conjunction with the ScanMask field in the gövt resource. If any of the 1 bits in a ship's government's ScanMask field match any of the 1 bits in a jünk type's ScanMask field, that government will consider that junk cargo type illegal. Set to zero if unused.
LCName	The lower-case string to display in the player-info dialog box, among other places, e.g. "machine parts".

Note: all outfits with identical, non-empty LCNames will be displayed together as one type in the player-info dialog.

Abbrev	The short string that is displayed in the player's status bar when the player is carrying jünk of this type, e.g. "Parts".
BuyOn	Control bit test expression. This jünk will only be available to be bought when this expression evaluates true. Leave blank if unused.
SellOn	Control bit test expression. This jünk will only be able to be sold when this expression evaluates true. Leave blank if unused.

The ïntf resource

The ïntf resource controls the appearance of the status bar by modifying the position and colour of the various status bar elements, as well as changing the status bar background image. Additionally, the use of multiple ïntf resources, along with proper values in the various gövt resources' Interface fields, allows the appearance of status bar to change based on what type of ship the player is piloting - this is a useless but fairly neat effect.

`BrightText`	Bright text colour.
`DimText`	Dim text colour.
`RadarArea`	Rectangular bounds of the radar display.
`BrightRadar`	Bright radar pixel colour.
`DimRadar`	Dim radar pixel colour (note that having an IFF outfit will override these colours).
`ShieldArea`	Rectangular bounds of the shield indicator.
`ShieldColor`	Shield bar colour.
`ArmorArea`	Rectangular bounds of the armour indicator.
`ArmorColor`	Armour bar colour.
`FuelArea`	Rectangular bounds of the fuel indicator.
`FuelFull`	Colour of the "full jumps" portion of the fuel indicator.
`FuelPartial`	Colour of the "partial fuel" portion of the fuel indicator.
`NavArea`	Rectangular bounds of the navigation display.
`WeapArea`	Rectangular bounds of the weapon display.
`TargArea`	Rectangular bounds of the target display.
`CargoArea`	Rectangular bounds of the cargo display.
`StatusFont`	Font to use for the status bar.
`StatFontSize`	Normal font size to use.
`SubtitleSize`	Font size for ship subtitles.
`StatusBkgnd`	ID of PICT resource to use as backdrop for status display. Values less than 128 are interpreted as 128.

Note that all colours in the ïntf resource are encoded the same as HTML colours, and that vertical shield/armour/fuel bars are properly handled by the game engine - to make an indicator bar vertical, set its height to be greater than its width. Also note that setting any of the above rectangles to have zero area (right=left, top=bottom) will cause the corresponding status indicator to not be drawn.

The mïsn resource

Missions are the crown jewel of the Nova datafile, as well as the largest and most complex resources in the game. Each misn resource corresponds to a single mission that the player can undertake, with the name of the mission (which the player sees in the mission list) being the name of the associated misn resource. The first six fields in a misn resource help Nova determine where and when the mission is available:

AvailStel	Which stellar objects (i.e. planets) the mission is available at.
-1	Any inhabited stellar.
128-2175	ID number of a specific stellar.
5000-7047	Stellar in a system adjacent to specific system.
9999-10255	Specific govt's stellar.
15000-15255	Specific govt's ally's stellar.
20000-20255	Stellar of anybody but this specific govt.
25000-25255	Specific govt's enemy's stellar.
30000-30255	Stellar of specific govt or any of its class mates.
31000-31255	Stellar not of specific govt nor of any of its class mates.

AvailLoc	Where on a planet this mission is available.
0	From the mission computer.
1	In the bar.
2	Offered from ship (must set up associated përs resource as well).
3	In the main spaceport dialog.
4	In the trading dialog.
5	In the shipyard dialog.
6	In the outfit dialog.

AvailRecord	What your legal record in this system must be for this mission to become available.
0	ignored.
pos. value	record must be at least this high.
neg. value	record must be at least this low.
-32000	when the player has dominated the stellar in question.
-32001	when the player has dominated at least one stellar.

AvailRating	What your combat rating must be for this mission to be available.
-1	ignored.
0+	rating must be at least this high.

AvailRandom	A randomization factor, to ensure that some missions aren't available all the time. Mission randomizing values are recalculated each time you warp into a system.
100	always available.
1-99	available this % of the time.

The next two fields in the misn resource define where the player needs to go to complete the mission:

TravelStel	Which stellar object the player must go to during the mission.
-1	No specific stellar destination.
-2	A random inhabited stellar.
-3	A random uninhabited planet.

`128-2175`	ID number of a specific stellar (note that the mission travel objectives will also be fulfilled when landing on a duplicate stellar that has the identical name and coordinates to the stellar you specify here).
`9999-10255`	Random stellar of a specific govt.
`15000-15255`	Random stellar of a specific govt's ally.
`20000-20255`	Random stellar of anybody but this specific govt.
`25000-25255`	Random stellar of specific govt's enemy.
`30000-30255`	Random stellar of specific govt or any of its class mates.
`31000-31255`	Random stellar not of specific govt nor of any of its class mates.

`ReturnStel`	Where the player must return to in order to complete the mission and receive payment.
`-1`	No specific stellar destination.
`-2`	A random inhabited stellar.
`-3`	A random uninhabited stellar.
`-4`	The initial stellar, where the mission was accepted.
`128-2175`	ID number of a specific stellar (note that the mission will also complete when landing on a duplicate stellar that has the identical name and coordinates to the stellar you specify here).
`9999-10255`	Random stellar of a specific govt.
`15000-15255`	Random stellar of a specific govt's ally.
`20000-20255`	Random stellar of anybody but this specific govt.
`25000-25255`	Random stellar of specific govt's enemy.
`30000-30255`	Random stellar of specific govt or any of its class mates.
`31000-31255`	Random stellar not of specific govt nor of any of its class mates.

The next five fields tell Nova about any special cargo associated with a mission:

`CargoType`	What type of cargo must be carried.
`-1`	No special cargo for this mission.
`0-255`	Specific cargo type.
`1000`	Random cargo of types 0-5 (the standard types).

`CargoQty`	What amount of cargo must be carried.
`-1`	Ignored (no cargo).
`0 and up`	This many tons of cargo.
`-2 and below`	abs(CargoQty) tons, +/- 50%.

`PickupMode`	Where the cargo is to be picked up.
`-1`	Ignored.
`0`	Pick up at mission start.
`1`	Pick up at TravelStel.
`2`	Pick up when boarding special ship.

`DropOffMode`	Where the cargo is to be dropped off.
`-1`	Ignored.
`0`	Drop off at TravelStel.
`1`	Drop off at mission end (ReturnStel).

Note that DropOffMode 1 will only cause cargo to be dropped off if the cargo was picked up previously, and then only if the mission's special ship goal has been completed or the mission has no special ship goal.

Note: don't set your cargo to be picked up and dropped off at the same place, as it may cause Nova to behave strangely.

ScanMask	Used to determine which governments consider your cargo illegal. If any of the 1 bits in this field match any of the 1 bits in a government's ScanMask field, that govt will consider this mission's cargo illegal. Set to zero if unused.

The next field tells Nova what to give you if you're successful in your mission:

PayVal	What you get if you're successful and you return to ReturnStel.
`0 or -1`	No pay.
`1 and up`	This number of credits.
`-10128 to -10383`	Clean legal record with the govt with this ID.
`-20128 to -20383`	Clean legal record with the govt with this ID, and all its allies.
`-30128 to -30383`	Clean legal record with the govt with this ID, and all its classmates.
`-40001 to -40099`	Take away this % of the player’s cash (-40001 = 1%, -40002 = 2%, etc.).
`-50000 and down`	Take away this number of credits at mission start (-50000 = 0, -50001 = 1, etc.).

The next six fields contain information on the special ships associated with this mission, if any:

ShipCount	The number of special ships for this mission.
`-1`	Ignored (no special ships).
`0-31`	This number of special ships.

ShipSyst	Which system the special ships will appear in.
`-1`	The initial system where the mission is begun.
`-2`	Any random system.
`-3`	TravelStel's system.
`-4`	ReturnStel's system.
`-5`	System adjacent to initial system.
`-6`	Whatever system the player is in (i.e. follow him around).
`128-2175`	ID number of a specific system.
`9999-10255`	Specific govt's system.
`15000-15255`	Specific govt's ally's system.
`20000-20255`	System of any govt but this specific one.
`25000-25255`	Specific govt's enemy's system.
`30000-30255`	System of a specific govt or any of its class mates.
`31000-31255`	System not of specific govt nor of any of its class mates.

ShipDude	What dude resource to use to determine the special ship's types and characteristics.
`-1`	Ignored (no special ships).
`128-639`	ID number of a specific dude class.

ShipGoal	The mission goal associated with the special ships.
`-1`	Ignored (no specific goal for the special ships).
`0`	Destroy all the ships.
`1`	Disable but don't destroy them.
`2`	Board them.
`3`	Escort them (keep them from getting killed).
`4`	Observe them (for ships that can cloak, at least one must be visible onscreen - for ships that cannot cloak, the player must merely be in the same system as them).
`5`	Rescue them (they start out disabled and stay that way until you board them; to make a rescue mission where the ship stays disabled, give the special ships a govt with the "always disabled" flag set).

6	Chase them off (either kill them or scare the into jumping out of the system).

ShipBehav	Defines any special actions you want the ships to take.
-1	Ignored (they use their standard AI routines).
0	Special ships will always attack the player.
1	Special ships will protect the player.
2	Special ships will attempt to destroy enemy stellars.

ShipNameID	Tells Nova how to name the special ships.
-1	Ignored (special ships have normal names).
128 and up	Pick a name from this STR# resource.

ShipStart	Defines where in the system the ships will start.
-4	Appear on top of nav default 4.
-3	Appear on top of nav default 3.
-2	Appear on top of nav default 2.
-1	Appear on top of nav default 1.
0	Appear randomly in the system (as usual).
1	Jump in from hyperspace after a short delay.
2	Appear randomly, cloaked.

Note that a ShipStart value of 0 (appear randomly in the system) is the proper value to use in conjunction with përs resource flag 0x0040.

CompGovt	Which government to use in determining how your record changes on completing this mission.
-1	Ignored (no reward other than pay).
128-383	Increase record with this govt.

CompReward	How much to increase your record with CompGovt.
(any value)	Increase record by this much.

Note: if you have a CompGovt and reward defined and you fail the mission, that govt will take it personally and decrease your record by 1/2 the amount specified in CompReward. This is useful for making missions whose success is considered vital by a certain party.

ShipSubtitle	Tells Nova which subtitle, if any, to use for the special ships.
-1	Ignored (special ships have normal subtitles).
128 and up	Pick a subtitle from this STR# resource.

BriefText	The desc to show in the dialog that comes up when you accept a mission. (formats are the same for all seven fields).
-1	No special mission briefing.
128 and up	ID number of the desc resource to use (ID numbers of 5000 and up are usually the safest).

QuickBrief	The desc to show when the user hits the "Mission Briefing" (I) key.
LoadCargText	The desc to show when special mission cargo is loaded from a planet.
DumpCargoText	The desc to show when special mission cargo is offloaded (not jettisoned into space as the name would suggest!).
CompText	The desc to show when you go to ReturnStel and the mission has been successful.

FailText	The desc to show when you go to ReturnStel and the mission has been a failure.

ShipDoneText	The desc to show when you complete the special ship goal.

CanAbort	
0	This mission can't be aborted by the player.
1	This mission can be aborted by the player.

TimeLimit	Like it says.
-1 or 0	Ignored (no time limit).
1 and up	This number of days.

The next few fields tell Nova about any auxiliary ships you want to be placed in the universe for this mission. Auxiliary ships cannot be given specific instructions, and no goals can be set for them; they simply are "normal" ships that are placed into the universe for the purpose of adding atmosphere to mission.

AuxShipCount	How many aux ships, if any, to activate for this mission:
-1	No aux ships.
1-31	Place this many aux ships in the universe.

AuxShipDude	ID number of the specific dude resource to use to set up the aux ships.

AuxShipSyst	What systems to place the aux ships in:
-1	Any system the player is in.
-2	TravelStel's system.
-3	ReturnStel's system.
128-2175	ID number of a specific system.
5000-7047	In this system, or any systems adjacent to it.
9999-10255	Any system belonging to this govt.
15000-15255	Any system belonging to this govt or its allies.
20000-20255	Any system not belonging to this govt.
25000-25255	Any system belonging to enemies of this govt.
30000-30255	Any system of a specific govt or any of its class mates.
31000-31255	Any system not of specific govt nor of any of its class mates.

Flags	Some misc. flag bits.
0x0001	Marks the mission as an auto-aborting mission, which will automatically abort itself after it is accepted. (sometimes useful to create special ships) Any control bits pointed to by the mission’s CompBitSet fields will be automatically set when the mission aborts.

Note: there must be special ships associated with the mission to trigger the auto-abort. If the mission is one in which a special ship replaces a përs ship at mission start (such as for a "rescue disabled ship" mission) and the SpecialShipGoal is 2 or 5 (board or rescue) the mission will auto-abort after the special ship is boarded.

0x0002	Don’t show the red destination arrows on the map.
0x0004	Can’t refuse the mission.
0x0008	Mission takes away 100 units of fuel upon auto-abort. (mission won't be offered if player has less than 100 units of fuel).
0x0010	Infinite auxShips.
0x0020	Mission fails if you're scanned.
0x0040	Apply -5x CompReward reversal on abort.

0x0080	Global penalty when jettisoning mission cargo in space (currently ignored).
0x0100	Show green arrow on map in initial briefing.
0x0200	Show an additional arrow on the map for the ShipSyst.
0x0400	Mission is invisible and won't appear in the mission info dialog. (be careful with this!).
0x0800	The special ships' type will be selected at mission start and then kept the same whenever the special ships for that mission are created, until the mission ends. This can be used for (e.g.) "attack pirate" missions where you want the type of the enemy ship to be random at first but you don't want it to change every time the player lands or re-enters the system.
0x2000	Mission unavailable if player's ship is of inherentAI type 1 or 2 (cargo ships).
0x4000	Mission unavailable if player's ship is of inherentAI type 3 or 4 (warships).
0x8000	Mission will fail if player is boarded by pirates.
Flags2	More flag bits.
0x0001	Don't offer mission if the player doesn't have enough cargo space to hold the mission cargo (even if the mission cargo won't be picked up until later).
0x0002	Apply mission Pay on auto-abort.
0x0004	Mission fails if player is disabled or destroyed.
AvailShipType	Defines what class of ship you must have for this mission to be made available:
0 or -1	Ignored.
128-255	Must be flying a ship of this type.
1128-1255	Must not be flying a ship of this type.
2128-2255	Must be flying a ship of this inherent govt.
3128-3255	Must not be flying a ship of this inherent govt.
RefuseText	The desc, if any, to show when mission offered in a bar or from a passing ship is refused.
AvailBits	A control bit test expression that determines the mission's availability. If not left blank, the mission will only be available when the expression evaluates to true.
OnAccept	Control bit set expression which is evaluated when the mission is accepted by the player.
OnRefuse	Control bit set expression which is evaluated when the mission is refused by the player.
OnSuccess	Control bit set expression which is evaluated when the mission is completed successfully.
OnFailure	Control bit set expression which is evaluated when the mission is failed.
OnAbort	Control bit set expression which is evaluated when the mission is aborted by the player.
OnShipDone	Control bit set expression which is evaluated when the mission's special ship goal is completed.
Require	These two Require fields together form a 64-bit flag that is
Require	logically and'ed with the Contribute fields from the player's current ship and outfit items. If for each 1 bit in the Require fields there is a matching 1 bit

	in one or more of the Contribute fields, the mission will be available. Leave these set to zero if unused.
DatePostInc	If this is set to something greater than zero, the game date will be advanced by this number of days after successful completion or auto-aborting of the mission.
AcceptButton	The label to use for the Accept button in the initial mission briefing dialog. If no text is entered here, Nova will use the "Yes" button label from STR# 150 if the mission can be refused, or the "Okay" button label if the mission is non-refusable.
RefuseButton	The label to use for the Refuse button in the initial mission briefing dialog, for missions that can be refused. If no text is entered here, Nova will use the "No" button label from STR# 150.
DispWeight	Controls the order that the mission is presented in the bar and mission BBS list. Missions with higher DispWeight values are presented first.

Whenever Nova displays a desc resource related to a mission, such as the initial mission description (desc ID 4000-4255) or one of the special mission briefings (e.g. CompText and QuickBrief) it performs one other special operation on the text. It searches through the text and replaces a few special "wildcard" symbols with pertinent mission information. This is extremely useful in setting up mission briefings that include random information that wouldn't be known when the description is written. These special symbols and their expansions are:

<DSY>	The name of the destination system.
<DST>	The name of the destination stellar.
<RSY>	The name of the return system.
<RST>	The name of the return stellar.
<CT>	The name of the type of cargo to be carried.
<CQ>	The quantity of cargo to be carried.
<DL>	The date of the mission deadline, if any.
<PAY>	Absolute value of mission pay (does nothing if the mission pay isn't monetary).
<REG>	Who Nova is registered to, or "UNREGISTERED".
<PN>	The player's name.
<PNN>	The player's nickname. If no nickname was specified, Nova will use the player's full name here instead.
<PSN>	The player's ship's name.
<PST>	The player's ship type (comes from the resource name of the player's ship type's shïp resource).
<PRK>	The ConvName of the highest-weighted active rank resource. If none is found, this will be replaced with "captain".
<SRK>	The ShortName of the highest-weighted active rank resource. If none is found, this will be replaced with "captain".
<PRKnnn>	Same as <PRK>, but only for ranks affiliated with government ID nnn. Note that you can only do this once per description, or bad things might happen.
<SRKnnn>	Same as <SRK>, but only for ranks affiliated with government ID nnn. Note that you can only do this once per description, or bad things might happen.
<RRK>	The full name of the most recently activated rank resource. Note that it's best to only use this in a mission briefing where you know that you've just given the player a rank, because otherwise bad things could happen. (e.g. the most recently activated rank pointer isn't cached between game sessions).
<OSN>	The offering ship name (only works when offering a mission from a ship).

<SN>	Special ship name (Note: Nova will screw up if you use this in the initial mission description, as it doesn't pick the special ship names until you actually accept the mission.).

Note: Mission cargo names that start with the asterisk character ('') are treated as "quantity less" - i.e. their quantity is never shown in the player-info dialog and the word "the" is omitted in the message that appears when the player retrieves that type of cargo from a ship as part of a mission. This is useful for when the mission cargo is a proper name.*

Note: When Nova selects a random mission destination, it attempts to ensure that the random destination is: a) more than two hyper jumps away from the system where the mission is being offered, and b) a stellar which is guaranteed to always exist throughout the course of the game, regardless of system swapping. This means that if you have multiple systems that occupy the same coordinates at different times in the game and not all of them contain stellar object X, no missions will use stellar object X as their random destination. (This is to keep the player from accepting a mission with a random destination that might cease to exist before he gets there!) A further important consequence of this restriction is that if Nova detects that a mission whose TravelStel or ReturnStel is to be randomly selected will violate this rule, it will be prevented from being offered regardless of any other availability parameters the mission might have - i.e. if you create a mission and Nova refuses to make it available, check the debug log to see if the mission couldn't find a suitable nontransient random stellar of the desired characteristics.

The nëbu resource

Nëbu resources contain info on the nebulae (or other space phenomena) which are displayed in the background of the star map. These images don’t actually have any effect on events in the game, they’re just there to look pretty. You can, however, combine nëbu background images with custom asteroid or interference data in the sÿst resources for cool localized effects, and you can also use the nëbu resource's OnExplore field to enable events when the player explores a certain nebula.

The PICT resources associated with the 32 available nëbu resources are numbered 9500-9724. Each nebula can have up to seven PICT resources associated with it; the first nebula's PICTs are 9500-96, the second's are 9507-13, etc. The engine will pick the best nebula image to display for a given map scale (the map scales used in Nova are 42.1%, 56.2%, 75.0%, 100.0%, 133.3%, 177.7%, and 237.0%). If nebula image of the proper size for the current map scale isn't available, Nova will pick the closest one and stretch it as necessary. PICTs for a given nebula should be sorted by size in ascending order.

The effects of the nëbu resource’s fields are as follows:

XPos **YPos**	The image’s position on the star map. These coordinates are expressed in the scale of the 'normal' map zoom level (not zoomed in or out) and are relative to the upper-left corner of the image.
XSize **YSize**	The image’s size on the star map. These values are expressed in the scale of the ‘normal’ map zoom level (not zoomed in or out) and tell Nova how big to make the image when the map is at normal zoom.
ActiveOn	Control bit test expression. Leave blank if unused.
OnExplore	Control bit set expression that is evaluated whenever the nebula is tested for visibility and found to be visible. Note that this field is handled a little differently from other control bit set expressions in Nova, because the explored state of the nebulae isn't saved from game to game; rather, it is recalculated every time based on the systems the player has explored. As a result, this expression may get evaluated multiple times, so you shouldn't use the random operator or any operators which do anything but set or clear control bits here.

The öops resource

Oops resources contain info on planetary disasters. Actually, the term 'disasters' is a misnomer, as these occurrences simply affect the price of a single commodity at a planet or station, for good or bad. Nova uses the name of the resource in the commodity exchange dialog box to indicate that a disaster is currently going on at a planet. The fields of an oops resource are:

`Stellar`	The stellar object this disaster is linked to.
`128-1628`	ID of a stellar object.
`-1`	Any planet or station (use sparingly).
`-2`	Nothing (used for mission-related news).
`Commodity`	Which commodity to affect the price of (0 = food, 1 = industrial, etc.).
`PriceDelta`	How much to raise or lower the price. (negative numbers lower it).
`Duration`	How many days the disaster lasts.
`Freq`	Percent chance per day that the disaster will occur.
`ActivateOn`	Control bit test expression. Leave blank if unused.

The oütf resource

Outf resources store information on the items that you can buy when you choose 'Outfit Ship' at a planet or station.

`DispWeight`	The display weight of this item. Items with a higher display weight are shown closer to the top of the outfit dialog. This can be used to effectively rearrange the order in which items are displayed without rearranging the resources themselves.
`Mass`	The mass in tons of the item (0 = no appreciable mass).
`TechLevel`	What the technology level of the item is. This item will be available at all spaceports with a tech level of this value or higher. (The exception to this rule involves the SpecialTech fields of the spöb resource; see the section on spöb resources for more information.)

The next two fields tell Nova what kind of modification this item performs:

If ModType is:	Then it’s:	And ModVal refers to:
1	a weapon	The ID number of the associated wëap resource
2	more cargo space	The number of tons of cargo space to add
3	ammunition	The ID number of the associated wëap resource
4	more shield capacity	The number of shield points to add
5	faster shield recharge	How much to speed up (1000 = one more shield point per frame)
6	armour	The number of armour points to add
7	acceleration booster	Amount of accel to add (see shïp for more info)
8	speed increase	Amount of speed to add (see shïp for more info)
9	turn rate change	Amount of turn change (100 = 30¡/sec)
10	unused	
11	escape pod	ignored
12	fuel capacity increase	Amount of extra fuel (100 = 1 jump)
13	density scanner	ignored
14	IFF (colorized radar)	ignored
15	afterburner	How much fuel to use (units/sec)
16	map	
	1 and up	How many jumps away from present system to explore
	-1	Explore all inhabited independent systems
	-1000 & down	Explore all systems of this govt class (-1000 is govt class 0, -1001 is govt class 1, etc.)
17	cloaking device	
	0x0001	Faster fading
	0x0002	Visible on radar
	0x0004	Immediately drops shields on activation
	0x0008	Cloak deactivates when ship takes damage
	0x0010	Use 1 unit of fuel per second
	0x0020	Use 2 units of fuel per sec
	0x0040	Use 4 units of fuel per sec
	0x0080	Use 8 units of fuel per sec
	0x0100	Use 1 unit of shield per sec
	0x0200	Use 2 units of shield per sec
	0x0400	Use 4 units of shield per sec
	0x0800	Use 8 units of shield per sec
	0x1000	Area cloak - ships in formation with a ship carrying this cloaking device will also be cloaked

18	fuel scoop	How many frames per 1 unit of fuel generated. Enter a negative value to perform the same function in 'fuel sucking' mode
19	auto-refueller	ignored
20	auto-eject	ignored (requires escape pod to work)
21	clean legal record	ID of govt to clear legal record with, or -1 for all
22	hyperspace speed mod	Number of days to increase or decrease ship’s hyperspace travel time (still can’t go below 1 day/jump).
23	hyperspace dist mod	Amount to increase or decrease the no-jump zone’s radius by (the standard radius is 1000) .
24	interference mod	Subtracts the value in ModVal from the current star system’s Interference value when calculating how 'fuzzy' to make the radar scanner.
25	marines	Adds the value in ModVal to your ship’s effective crew complement when calculating capture odds.
	1 and up	Add this number to the player's ship's effective crew size
	-1 to -100	Increase the player's capture odds by this amount (e.g. -5 is an increase of 5%).
26	(ignored)	
27	increase maximum	The ID number of another outfit item, (call it "B") whose maximum value is to be increased. Item B’s standard maximum will be multiplied by the number of items the player has that have a ModType of 27 and point to B. If the player owns no items that modify the maximum of item B, its maximum will be unchanged.
28	murk modifier	The amount by which to increase or decrease the current system's murkiness level.
29	faster armour recharge	How much to speed up (1000 = one more armour point per frame).
30	cloak scanner	
	0x0001	reveal cloaked ships on radar.
	0x0002	reveal cloaked ships on the screen.
	0x0004	allow targeting of untargetable ships.
	0x0008	allow targeting of cloaked ships.
31	mining scoop	
32	multi-jump	number of extra jumps to perform when the user initiates a hyperspace jump.
33	Jamming Type 1	amount of jamming to add or subtract
34	Jamming Type 2	amount of jamming to add or subtract
35	Jamming Type 3	amount of jamming to add or subtract
36	Jamming Type 4	amount of jamming to add or subtract
37	fast jumping	(grants carrying ship the ability to enter hyperspace without slowing down).
38	inertial dampener	(makes ship inertialess)
39	ion dissipater	amount of deionization to add. 100 equals 1 point of ion energy per 1/30th of a second. Higher values yield faster ion charge dissipation.
40	ion absorber	amount of extra ionization capacity to add
41	gravity resistance	
42	resist deadly stellars	
43	paint	the colour to paint the player's ship, encoded as a 15-bit colour value, where the bits are: 0RRRRRGGGGGBBBBB (this is necessary because the ModVal field isn't big enough to hold a 32-bit HTML colour value).
44	reinf. Inhibitor	a govt class value. any govt with this value in its Class1-4 fields will be prevented from calling in reinforcements while the player is in the system and has this outfit. setting this field to -1 will inhibit reinforcements for all ships regardless of govt. note that this outfit will only work when carried by the player.
45	modify max guns	add/subtract from max number of guns
46	modify max turrets	add/subtract from max number of turrets
47	bomb	destroys the player in flight. set the modval to the ID of the desc resource to show after the player is destroyed, or -1 for none.

48	IFF scrambler	a govt class value. any govt with this value in its Class1-4 fields will fooled into thinking the player is a friendly ship and will not attack without provocation. note that this outfit will only work when carried by the player.
49	repair system	will occasionally repair the ship when it's disabled
50	nonlethal bomb	randomly destroys itself and damages the player (nonfatally) in flight. set this field to the ID of a bööm resource to display when an item of this type self-destructs

The next two fields tell Nova how many of this item you can possibly have at once:

Max	How many you can have (not counting weapon limitations).

Flags	Miscellaneous info:
0x0001	This item is a fixed gun.
0x0002	This item is a turret.
0x0004	This item stays with you when you trade ships (persistent).
0x0008	This item can't be sold.
0x0010	Remove any items of this type after purchase (useful for permits and other intangible purchases).
0x0020	This item is persistent in the case where the player's ship is changed by a mission set operator. The item's normal persistence for when the player buys or captures a new ship is still controlled by the 0x0004 bit.
0x0100	Don't show this item unless the player meets the Require bits, or already has at least one of it.
0x0200	This item's total price is proportional to the player's ship's mass. (ship class Mass field is multiplied by this item's Cost field)
0x0400	This item's total mass (at purchase) is proportional to the player's ship's mass. (ship class Mass field is multiplied by this item's Mass field and then divided by 100) Only works for positive-mass items.
0x0800	This item can be sold anywhere, regardless of tech level, requirements, or mission bits.
0x1000	When this item is available for sale, it prevents all higher-numbered items with equal DispWeight from being made available for sale at the same time.
0x2000	This outfit appears in the Ranks section of the player info dialog instead of in the Extras section.
0x4000	Don't show this item unless its Availability evaluates to true, or if the player already has at least one of it.

The next field, Cost, tells Nova how much to charge you for the item.

The next few fields (ModType2-4 and ModVal2-4) are for you to specify 'alternate' functions for an outfit item - e.g., a weapon could also reduce the ship's turning speed. The only restriction on ModType2-4 is that you shouldn't use it for weapons or ammo (modtypes 1 or 3).

Availability	Control bit test expression. Leave blank if unused. Note that depending on the configuration of other flags, the item might appear in the outfit window even if Availability is false (it will still not be able to be purchased)
OnPurchase	Control bit set expression. Leave blank if unused.
Contribute Contribute	These two Contribute fields together form a 64-bit flag that is subsequently combined with the Contribute fields from the player's ship and the other outfit items in the player's possession, to be used with the Require fields in the outf and misn resources.
Require	These two Require fields together form a 64-bit flag that is

Require	logically and'ed with the Contribute fields from the player's current ship and outfit items. If for each 1 bit in the Require fields there is a matching 1 bit in one or more of the Contribute fields, the item will be available. Leave these set to zero if unused. Note that depending on the configuration of other flags, the item might still appear in the outfit window even if the player doesn't meet the Require bits (it will still not be able to be purchased).
OnSell	Evaluated when the item is sold.
ItemClass	The item's classification, used in the përs resource for items that are given out by non-player characters' ships. Set to 0 or -1 if unused.
ScanMask	If this is an "illegal" outfit type, this is used in conjunction with the ScanMask field in the gövt resource. If any of the 1 bits in a ship's government's ScanMask field match any of the 1 bits in an oütf type's ScanMask field, that government will consider that outfit item illegal. Set to zero if unused.
BuyRandom	The percent chance that an item of this type will be available for purchase on a given day, from 1-100. Values less than 1 or greater than 100 are interpreted as 100.
ShortName	The short string that is displayed in the outfit dialog menu for this item type. If you want to split this name into two separate lines, put the characters "\n" into the name, e.g.: "Real Big\nScary Gun". When using this, lines that start with an alphanumeric character are drawn in white, while lines that start with other symbols are drawn in grey.
LCName	The lower-case, singular name that's displayed in the player-info dialog and other places, e.g. "real big scary gun".
LCPlural	The lower-case, plural name that's displayed in the player-info dialog and other places, e.g. "real big scary guns".
RequireGovt	Which governments the outfit item's Require bits pertain to:
-1	Requirements apply in all outfit shops.
128-383	Requirements apply only on stellars belonging to this govt or its allies.
1128-1383	Requirements apply only on independent stellars and stellars belonging to this govt or its allies.
2128-2383	Requirements apply on all stellars except those belonging to this govt or its allies.
3128-3383	Requirements apply on all stellars except independent stellars or stellars belonging to this govt or its allies.

The përs resource

The pers resource defines the characteristics of an AI personality – that is, a specific person the player can encounter in the game. These AI-people have their names (which are also the names of the associated përs resource) displayed on the target-info display in place of the name of their ship class. When ships are created, there is a 5% chance that a specific AI-person will also be created. (obviously, as AI-people are killed off, they cease to appear in the game.) The first field tells Nova where a certain person can be encountered:

`LinkSyst`	Which systems the person can be created in.
`-1`	Any system.
`128-2175`	ID of a specific system.
`9999-10255`	Any system belonging to this specific government.
`15000-15255`	Any system belonging to an ally of this govt.
`20000-20255`	Any system belonging to any but this govt.
`25000-25255`	Any system belonging to an enemy of this govt.

The next four fields define the person's character traits:

`Govt`	The person's governmental affiliation.
`-1`	Ignored (person is independent).
`128-383`	ID of a specific government.
`AI Type`	The person's AI type (see the section on düde resources).
`1`	Wimpy trader.
`2`	Brave trader.
`3`	Warship.
`4`	Interceptor.
`Aggress`	The person's aggression, i.e. how close ships have to be before the person will attack them, on a scale of 1 (close) to 3 (far).
`Coward`	At what percent of total shield capacity will the person run away from a fight? e.g. a value of 25 would cause the person to retreat when his shields dropped to 25%.

The next fields tell Nova more about the ship that a person uses:

`ShipType`	ID number of the person's ship class.
`WeapType` `(x8)`	ID numbers of weapon types.
`-1 or 0`	No weapon.
`128-383`	Add this weapon type.
`WeapCount` `(x8)`	How many of each of the above weapons to add (Note: This is in addition to the standard weapons already included with the ship. Standard weapons can be "removed" by entering their ID numbers in the WeapType fields and entering the negative of their standard load for the given ship class in the WeapCount field.)
`-1 or 0`	None.
`1 and up`	Add this many.
`AmmoLoad` `(x8)`	The standard ammo load for weapons that need it, or ignored for those that don't
`-1 or 0`	No ammo.

`1 and up`	Include this many rounds of ammo.

`Credits`	How many credits the person carries.
`0`	ignored (no credits).
`1 and up`	This many credits, +/- 25%.

`ShieldMod`	How much to increase/decrease the person's shield capacity, in percent. For example, a value of 130 entered here would make the person's ship have shields that are 30% stronger than if he were flying a stock ship. Similarly, a value of 70 would make his shields 30% weaker. A value less than zero makes this person invincible.

The next fields tell Nova about any special quotes or missions to link to this ship:

`HailPict`	ID number of a PICT resource to be displayed in the communications dialog instead of the standard picture for this person's ship type.
`CommQuote`	Index number of an entry in STR# resource 7100, to be displayed in the communications dialog.
`HailQuote`	Index number of an entry in STR# resource 7101, to be displayed at the bottom of the game screen (i.e. over the radio).
`LinkMission`	What mission to activate when the ship is boarded or hailed.

`Flags`	Some control bits.
`0x0001`	The special ship will hold a grudge if attacked, and will subsequently attack the player wherever the twain shall meet.
`0x0002`	Uses escape pod & has afterburner.
`0x0004`	HailQuote only shown when ship has a grudge against the player.
`0x0008`	HailQuote only shown when ship likes player.
`0x0010`	Only show HailQuote when ship begins to attack the player.
`0x0020`	Only show HailQuote when ship is disabled.
`0x0040`	When LinkMission is accepted with a single SpecialShip, replace it with this ship while removing this one from play. This is generally only useful for escort and refuel-a-ship missions.

Note: if the mission's SpecialShip düde type contains the përs ship's ship type in it, the SpecialShip that's created will be of the same type as the përs ship, regardless of the probabilities in the düde resource. This is to prevent a përs ship from accidentally morphing into another ship type before the player's eyes. If you really do want to have the përs ship be replaced by a SpecialShip of a different type, use a düde in the mission's ShipDude field that doesn't contain the përs's ship type.

`0x0080`	Only show quote once.
`0x0100`	Deactivate ship (i.e. don’t make it show up again) after accepting its LinkMission.
`0x0200`	Offer ship’s LinkMission when boarding it instead of when hailing it.
`0x0400`	Don’t show quote when ship’s LinkMission is not available.
`0x0800`	Make ship leave after accepting its LinkMission.
`0x1000`	Don’t offer if player is flying a wimpy freighter (aiType 1).
`0x2000`	Don’t offer if player is flying a beefy freighter (aiType 2).
`0x4000`	Don’t offer if player is flying a warship (aiType 3).
`0x8000`	Show disaster info when hailing.

`ActiveOn`	Control bit test expression. Leave blank if unused.

GrantClass	The class of outfit item that is given out by this person's ship when boarded by the player. If there are multiple outfit items with the same ItemClass value, Nova will choose a random outfit item of that ItemClass. Set to 0 or -1 if unused.
GrantProb	The probability that this person will grant the player any items when boarded. Set to 100 for maximum chance, zero if unused.
GrantCount	The maximum number of outfit items to be given when boarded by the player - the actual value given will be between GrantCount/2 and GrantCount.
Colour	The colour to paint this person's ship, encoded the same as HTML colours (00RRGGBB). Set to 00000000 if unused (which is the same as 00FFFFFF, or "pure white paint" i.e. no shading).
Flags2	
0x0001	This person starts with zero fuel.

The ränk resource

The rank resource is used to give the player a feeling of 'belonging' to a given government. It can also be used to give the player certain advantages that come with rank. When a rank is made active (which is accomplished through any suitable control bit set string) the player is given all the privileges of that rank, whatever they might be, and the name of that rank is displayed in the player-info dialog.

Weight	The importance of this rank, relative to the other rank resources that might be active. Ranks with higher weight are displayed first in the player-info dialog, and the active rank with the highest weight is selected for the <PRK> and <PSR> mission briefing tags.
AffilGovt	The ID of the government affiliated with this rank.
Contribute	Another 64 bits of Contribute values that kick in when the rank is active. These can be used to prevent the player from buying certain items or doing certain missions until achieving a certain rank, for example.
Salary	The number of credits that the affiliated government will pay the player, per day.
SalaryCap	The maximum amount of money the player can have before the affiliated government stops paying the salary. Set to 0 or -1 if unused.
Flags	
0x0001	Deactivate all other active ranks affiliated with this same govt when this rank is activated (excludes permanent ranks).
0x0002	Deactivate all other active ranks affiliated with this same govt when this rank is deactivated (excludes permanent ranks).
0x0004	Deactivate this rank if player destroys or disables a ship of the affiliated government or its allies.
0x0008	Rank is permanent and cannot be deactivated except if explicitly done by a control bit eval string.
0x0010	Deactivate all other active and lower-weighted ranks affiliated with this same govt when this rank is activated (excludes permanent ranks).
0x0020	Deactivate all other active and lower-weighted ranks affiliated with this same govt when this rank is deactivated (excludes permanent ranks).
0x0040	Deactivate this rank if the player commits any crime against the affiliated government.
0x0100	Ships of the affiliated government will not automatically attack the player when he has this rank.
0x0200	All planets of the affiliated government will let the player land when he has this rank, regardless of their MinStatus field.
0x0400	Player can always request battle assistance from ships of the affiliated government, who will also call in reinforcements on the player's behalf if they are available.
0x0800	Ships allied with the affiliated govt will always repair or refuel the player for free.
PriceMod	Used to modify the prices of items and ships at planets owned by the affiliated government. A value of 100 equals 100% of original price (i.e. prices are unchanged). Higher or lower values raise or lower the prices correspondingly. (can be used to let distinguished players get special deals on ships and items at "friendly" planets that have granted them the rank).

The name of the ränk resource is the full name of the rank, displayed in the player-info dialog. example: "Commission of Space Marshall in the Hector Empire".

The text fields in the ränk resource are:

ConvName	The rank name as used in conversation, through mission briefings and the <PRK> tag. If this is set to an empty string, the rank will never be used in conversation. If there are no active ranks or none of the active ranks have ConvNames, the <PRK> tag will simply display "captain". example: if ConvName is "Space Marshall", then "Good job, <PRK>" becomes either "Good job, Space Marshall" or "Good job, captain".
ShortName	The short rank named as used in conversation through mission briefings and the <PSR> tag. Behaviour with an empty string is the same as for the ConvName field. example: if ShortName is "Marshall" then "Hi there, <PSR>" becomes either "Hi there, Marshall" or "Hi there, captain".

The röid resource

Nova supports up to 16 asteroid types, each of which can have its own special properties. röid resources 128-143 store the attributes for each asteroid type:

Strength	The strength of this asteroid type - this is equivalent to armour values for ships.
SpinRate	The frame advance rate of this asteroid type. A value of 100 is 30 frames per second (quite fast) - lower numbers are slower.
YieldType	Defines what the resource-boxes which are ejected from this asteroid type will contain:
0-5	This type of standard cargo.
1000-1127	This type of jünk resource.
YieldQty	The average resource yield of this type of asteroid. Zero will cause asteroids of this type to not spit out any resource-boxes when destroyed; values of one or more will cause the asteroids to spit out approximately that number of resource-boxes (+/- 50%) when destroyed. Each resource-box is worth 1 unit of whatever cargo is defined in the YieldType field to be granted when a scoop-equipped ship runs over it.
PartCount	The number of particles that are thrown off when an asteroid of this type is destroyed.
PartColor	The colour of the particles, encoded the same as HTML colours. (00RRGGBB)
FragType1&2	Asteroids can break up into some number of sub-asteroids when destroyed. If both of these fields are used, Nova will randomly pick between them for each sub-asteroid that is created. If only one is used, all sub-asteroids will be of that type.
128-143	Generate this type of sub-asteroid.
-1	No sub-asteroids.
FragCount	The average number of sub-asteroids (+/- 50%) to be generated when an asteroid of this type is destroyed (zero for none).
ExplodeType	Type of explosion to show (0-63) when an asteroid of this type is destroyed. You can also add 1000 to the value of this field in the same manner as the ExplodeType field in the wëap resource. Set to -1 to not show any explosion.
Mass	Mass of this asteroid type (used when weapons hit asteroids).

The shïp resource

Spaceships are the heart of Nova, so the ship resource contains a lot of info. The name of a ship class, which is seen in the targeting display, corresponds to the name of the ship resource. The first nine fields give Nova some general performance info on each ship type:

Holds	Cargo capacity, in tons. Put a negative sign in front of this value if you want to prevent the player from purchasing mass expansions. (e.g. a value of -100 would mean 100 tons of hold space but no mass expansions allowed).
Shield	Shield strength.
Accel	Acceleration magnitude. 300 is considered an average value.
Speed	Top speed. 300 is also an average value here.
Maneuver	Turn rate. 10 Å 30¡/sec.
Fuel	Fuel capacity. 100 = 1 jump.
FreeMass	Space available to add additional items and upgrades. Note that this is in addition to the space taken up by the ship's stock weapons. (e.g. a ship with 20 tons listed in FreeMass and 10 tons of stock weapons will actually have 30 tons of expansion space, with 20 available.)
Armour	Armour strength.
ShieldRech	Shield recharge speed, in number of shield points x1000 per frame. Bigger numbers here mean faster recharge - a value of 1000 is equal to 1 point per frame or 30 points per second.

The next twelve fields tell Nova which stock weapons to put on your ship when you first buy it:

WeapType (x8)	ID numbers of weapon types.
-1 or 0	No weapon.
128-191	Add this weapon type.
WeapCount (x8)	How many of each of the above weapons to add.
-1 or 0	None.
1 and up	Add this many.
AmmoLoad (x8	The standard ammo load for weapons that need it, or ignored for those that don't.
-1 or 0	No ammo.
1 and up	Include this many rounds of ammo.

The next two fields tell Nova what this ship's maximum load out of fixed guns and turreted weapons is. Each ship has an inherent upper limit on fixed guns and turrets, in order to keep them from becoming absurdly powerful. (e.g. a bulk freighter has lots of room to add weapons, but is limited to a single turret for defence) The fields are:

MaxGun	The ship's maximum number of fixed guns, which are flagged in the WeapFlag field of the outf resource.

MaxTur	The ship's maximum number of turrets, which are flagged in the WeapFlag field of the outf resource.

The next field tells Nova where this ship is available for purchase:

TechLevel	What the technology level of the ship is. This ship will be available at all shipyards with a tech level of this value or higher. (The exception to this rule involves the SpecialTech fields of the spöb resource; see the section on spöb resources for more information.)

The next field, **Cost**, tells Nova how much to charge you when you buy this ship. The cost of buying a ship is always the cost of the new ship minus 25% of the original cost of your current ship and upgrades. (i.e. you always "trade up" to a new ship)

The next field stores info on how the ship explodes:

DeathDelay	The number of frames the ship "disintegrates" before finally exploding.
`0-59`	The ship disintegrates for this number of frames and then disappears in a single fireball.
`60+`	The ship disintegrates for this number of frames and then disappears in a huge explosion. The exact size of the resulting fireball is proportional to the ship's mass. (see below)

ArmorRech	Armour recharge speed, in number of armour points x1000 per frame. Bigger numbers here mean faster recharge - a value of 1000 is equal to 1 point per frame or 30 points per second.
Explode1	Type of explosion to show (0-63) while the ship is breaking up, or -1 to not show any explosions until the ship is finished being destroyed.
Explode2	Type of explosion to show (0-63) when the ship is completely destroyed. You can also add 1000 to the value of this field in the same manner as the ExplodeType field in the wëap resource. Set to -1 to not show any explosion when the ship is destroyed.
DispWeight	The display weight of this ship type. Ships with a higher display weight are shown closer to the top of the shipyard dialog. This can be used to effectively rearrange the order in which ships are displayed without rearranging the resources themselves.
Mass	The mass of the ship, in tons. This doesn't affect acceleration or speed at all, but it does affect travel time in hyperspace and the display on the density scanner. Also, the blast radius and impact strength when the ship explodes is proportional to its mass.
`1-99`	1 day per jump, small blip on density scanner.
`100-199`	2 days per jump, large blip on density scanner.
`200 and up`	3 days per jump, large blip on density scanner.

Length	The ship's length in meters. Currently unused in any calculations, but it's kinda cool, so it's displayed in the "detailed ship info" dialog.

The next field tells Nova what kind of AI the ship will have if it's not created in connection with a dude resource. The only place this field is useful is when a ship is created as an escort ship; otherwise, it's ignored:

InherentAI	What AI the ship uses when it's escorting the player.
1-4	Use this kind of AI. (see the AI descriptions above) Note that only ships with inherent AI of 1 or 2 can be used to carry cargo when they are the player's escorts.

Crew	Number of crew members. Ships with 0 crew can't be boarded, nor can they capture any other ships.

Strength	An arbitrary value that represents the relative strength of the ship type with respect to the rest of the universe. Used when calculating combat odds (see the govt resource for details).

The next field tells Nova what government is associated with a ship type.

Note that unlike previous EV games, Nova actually handles two inherent government associations for each ship type: an inherent combat govt (used when an AI ship or stellar is deciding if it likes or hates another ship) and an inherent attributes govt (used for non-combat things like voice type, distress message flags, etc, as well as for inherent jamming ability). Sometimes you might want to create a ship type that inherits attributes from a particular govt but isn't considered to be inherently of that govt in combat, so Nova lets you use the InherentGovt field in several different ways.

InherentGovt	
-1	No inherent combat govt or inherent attributes govt for this ship.
128-383	Ship is treated as being inherently of the govt with this ID, both for AI combat and attributes inheritance).
1128-1383	Ship has an inherent attributes govt with this ID (minus 1000) but no inherent combat govt.
2128-2383	Ship has an inherent combat govt with this ID (minus 2000) but no inherent attributes govt.

The next field is for some miscellaneous flags:

Flags	
0x0001	Slow jumping (75% normal speed).
0x0002	Semi-fast jumping (125%)
0x0004	Fast jumping (150%)
0x0008	Player ship takes advantage of FuelRegen property
0x0010	Ship is disabled at 10% armour instead of 33%
0x0020	Ship has afterburner when player has an advanced combat rating
0x0040	Ship always has afterburner (for AIs only)
0x0100	Show % armour on target display instead of 'Shields Down'
0x0200	Don't show armour or shield state on status display
0x0400	Ship is a planet-type ship, and can only be hit by planet-type weapons
0x1000	Ship’s turrets have a blind spot to the front
0x2000	Ship’s turrets have a blind spot to the sides
0x4000	Ship’s turrets have a blind spot to the rear
0x8000	Ship is an escape ship type - if the player is carrying any ships of this type and decides to eject, he will fly off in a ship of this type (with random damage) instead of an escape pod.

PodCount	For decorative purposes, AI ships can be made to launch escape pods when they're destroyed. This field contains the standard number of escape pods for an AI ship of this type to launch when destroyed, at a rate of one per second. Note that this has nothing to do with the përs field's escape pod flag, it's just for eye candy. Don't overuse this field, as it can be annoying if used too often. (perhaps restrict it only to luxury liner type ships).

DefaultItems	Up to eight default items with which to equip this ship when the player buys or captures one. Note that AI-controlled ships will ignore these fields; also, don’t put anything in here that isn’t a physical item - i.e. afterburners, shield boosters, and the like are okay, but no fake IDs, maps, etc.
128-255	Ship comes stock with this item.
-1	Ignored.
ItemCount	The number of each DefaultItem with which to equip the player.
FuelRegen	This ship type’s inherent fuel regeneration property. Works exactly the same as the fuel scoop ModVal property – useful for making ships with built-in fuel replenishment. Note that for the player to be able to use this field, the 0x0008 flag must also be set. (this allows you to give enemy ships built-in fuel scoops but still make the player have to buy his own)
SkillVar	The amount (in percent) to which this ship’s pilot’s skill varies. This affects acceleration and turn rate for each ship: i.e. a skill variance of 10% would make each ship of a given type up to 10% slower or faster than 'stock'. Values from 1 to 50% are valid.
Flags2	
0x0001	Ship exhibits swarming behaviour.
0x0002	Ship prefers standoff attacks.
0x0004	Ship can't be targeted.
0x0008	Ship can be fired on by point defence systems.
0x0010	Don't use fighter voices.
0x0020	Ship can jump without slowing down.
0x0040	Ship is inertia less.
0x0080	AI ships of this type will run away/dock if out of ammo for all ammo-using weapons.
0x0100	AI ships of this type will cloak when their weapon goes into burst reload.
0x0200	AI ships will cloak when running away.
0x0400	AI ships will cloak when hyper spacing.
0x0800	AI ships will cloak when just flying around.
0x1000	AI ships will not uncloak until close to their target.
0x2000	AI ships will cloak when docking.
0x4000	AI ships will cloak when pre-emptively attacked .
Availability	Control bit test expression. The player will be able to purchase this type of ship when the expression evaluates to true. Leave blank if unused. Depending on the configuration of other flags, the ship might appear in the shipyard but not be able to be purchased if its Availability evaluates to false.
AppearOn	Control bit test expression. Ships of this type will not show up in dude resources if this expression evaluates to false. Leave blank if unused.
OnPurchase	Control bit set expression. Leave blank if unused.
Deionize	The rate at which this ship type dissipates ionization charge. A value of 100 equals 1 point of ion energy per 1/30th of a second. Higher values for Deionise yield faster ion charge dissipation.

IonizeMax	The amount of ion charge at which a ship of this type will be considered "fully ionized".
KeyCarried	The key carried ship type, used for interesting effects in the wëap and shän resources.
DefaultItms2 **ItemCount**	More default items, used as above
Contribute Contribute	These two Contribute fields together form a 64-bit flag that is subsequently combined with the Contribute fields from the outfit items in the player's possession, to be used with the Require fields in the outf and misn resources.
Require Require	These two Require fields together form a 64-bit flag that is logically and'ed with the Contribute fields from the player's current ship and outfit items. If for each 1 bit in the Require fields there is a matching 1 bit in one or more of the Contribute fields, the ship can be purchased. Leave these set to zero if unused. Depending on the configuration of other flags, the ship might appear in the shipyard but not be able to be purchased if the player doesn't meet the Requirements.
BuyRandom	The percent chance that a ship of this type will be available for purchase on a given day. A BuyRandom of 0 means this ship will never be made available for purchase.
HireRandom	The percent chance that a ship of this type will be available for hire in the bar on a given day. A HireRandom of 0 means this ship will never be made available for hire.
OnCapture	Control bit set expression, evaluated when you capture a ship of this type. Leave blank if unused.
OnRetire	Control bit set expression, evaluated when you sell a ship of this type and/or replace it with a captured ship.
Subtitle	The subtitle to show on the target display for this ship type.
Flags3	Even more flags!
0x0001	Ship destroys asteroids.
0x0002	Ship scoops asteroid debris.
0x0010	Ship ignores gravity.
0x0020	Ship ignores deadly stellars.
0x0040	Ship's turreted shots appear above the ship instead of below.
0x0100	Don't show ship in shipyard if Availability is false.
0x0200	Don't show ship in shipyard if Require bits not met.
0x4000	When this ship is available for sale, it prevents all higher-numbered ship types with equal DispWeight from being made available for sale at the same time.
UpgradeTo	If an escort ship of this type can be upgraded, this field holds the ID of the ship type that it can be upgraded to. Set to 0 or -1 if this ship class can't be upgraded.
EscUpgrdCost	The cost to upgrade an escort ship of this type to the next more advanced version, as defined in the UpgradeTo field.

EscSellValue	The amount of cash the player gets for selling off a captured escort of this type. If you input a number that's less than or equal to zero here, Nova will default to 10% of the ship's original cost.
EscortType	Tells Nova which of the four categories of escorts to put this ship type into when organizing the escort control menu.
-1	Have the game try to figure it out at runtime.
0	Fighter.
1	Medium Ship.
2	Warship.
3	Freighter.
ShortName	The short string that is displayed in the shipyard dialog menu for this ship type. If you want to split this name into two separate lines, put the characters "\n" into the name, e.g.: "Big Ship\n(used)". When using this, lines that start with an alphanumeric character are drawn in white, while lines that start with other symbols are drawn in grey.
CommName	The short string to display for this ship when it is hailed by the player.
Long Name	The long string to display when the player purchases a ship of this type or starts a new pilot.
MovieFile	The filename of a QuickTime movie to display in place of the ship picture in the shipyard dialog. The QuickTime movie must reside within the Nova Files or Nova Plug-Ins folders and will be looped continuously while the player has this ship type selected.

Ships' target info picts are stored in PICT resources 3000 and on. The engine is smart enough to reuse targeting picts for two ship classes that have the same base sprites: all you have to do is give the first of any series of identical-looking ship types a target pict in the usual way (PICT resource ID 3000 + shipID - 128) and the engine will use it for all higher-numbered ship types with the same base sprites.

The spöb resource

Spob resources describe stellar objects, such as planets and space stations. (spob stands for space object) Each spob resource represents one stellar object, whose name is the name as the name of the resource. The first three fields tell Nova where to put the stellar and what graphics to use for it:

xPos & **yPos**	The stellar's X and Y positions in the system (0, 0) is centred.
Type	Which graphic to use, from 0 to 255.

The next field stores the flag bits that tell Nova what many of the characteristics of the stellar are. Perform an OR operation on the following **`flags`** to get the final flag value:

0x00000001	Can land/dock here.
0x00000002	Has commodity exchange.
0x00000004	Can outfit ship here.
0x00000008	Can buy ships here.
0x00000010	Stellar is a station instead of a planet.
0x00000020	Stellar is uninhabited (no traffic control or refuelling).
0x00000040	Has bar.
0x00000080	Can only land here if stellar is destroyed first.
0x00000000	Won't trade in food.
0x10000000	Low food prices.
0x20000000	Medium food prices.
0x40000000	High food prices.
0x00000000	Won't trade in industrial goods.
0x01000000	Low industrial prices.
0x02000000	Medium industrial prices.
0x04000000	High industrial prices.
0x00000000	Won't trade in medical supplies.
0x00100000	Low medical prices.
0x00200000	Medium medical prices.
0x00400000	High medical prices.
0x00000000	Won't trade in luxury goods.
0x00010000	Low luxury prices.
0x00020000	Medium luxury prices.
0x00040000	High luxury prices.
0x00000000	Won't trade in metal.
0x00001000	Low metal prices.
0x00002000	Medium metal prices.
0x00004000	High metal prices.
0x00000000	Won't trade in equipment.
0x00000100	Low equipment prices.
0x00000200	Medium equipment prices.
0x00000400	High equipment prices.

Tribute	The stellar's tribute payout when dominated.
-1 or 0	Default amount (1000 credits x Tech Level).
1 and up	This number of credits per day.

The next fields tell Nova what items and ships are available for purchase at this stellar:

TechLevel	What the base tech level of the stellar is. Only items and ships with TechLevels at or below this value will be available.
SpecialTech (x8)	Holds the special tech levels of this stellar. Unlike the previous field, only items and ships with exactly this TechLevel will appear here. This is useful for making low-tech worlds that also have a few high-tech items, or for flagging an item with an absurdly high TechLevel (say 15000) and then setting one of the SpecialTech fields of a particular stellar to that same value, thus making that item appear at that stellar and nowhere else.

The next two fields contain info on the stellar's governmental affiliation:

Govt	What government controls this stellar.
`-1`	ignored (stellar is independent).
`128-383`	number of the stellar's government.

MinStatus	The point on your record in the current system that you'll be denied landing clearance on this stellar.
`-32767`	Ignored (player can always land).
`-1 to -32766`	You can be this evil before they shun you.
`0 to 32766`	They have to like you this much before they let you land.
`32767`	Player can never land. (Note that this field is ignored if the stellar is uninhabited).

The next pair of fields tells Nova which special landscape to show and which ambient sound to play.

CustPicID	Which custom landscape to show.
`128 and up`	ID number of PICT to load instead of the standard landscape display.
`less than 128`	No custom landscape.

Note: for animated hypergates, this field can be optionally used to set the index of the transition between the "opening/closing" animation and the "working" animation. Set to 0 to have the engine use the first half of the frames for "opening/closing" and the second half for "working".

CustSndID	Which ambient sound to play.
`-1`	No ambient sound effect.
`Anything else`	ID number of snd resource to load.

Note: for hypergates and wormholes, this field serves a different purpose - it controls the angle at which ships emerge. Values between 0 and 359 specify an exact angle, while any other value specifies a random direction.;

The next two fields tell Nova what kind of ships, if any, to create for the planet's defence fleet:

DefenseDude	Which type of dude to use for the defence fleet:
`-1`	Ignored (no defence ships).
`128-639`	ID number of the dude resource to use to determine the defence ships' characteristics.

DefCount	The number of ships in the defence fleet. If you set this number to be above 1000, ships will be launched from the planet or station in waves. The last number in this field is the number of ships in each wave, and the first 3-4 numbers (minus 1 from the first digit) are the total number of ships in the planet's fleet. For example, a value of 1082 would be four waves of two

ships for a total of eight. A value of 2005 would create waves of five ships each, with 100 ships total in the planet's defence fleet.

Flags2

0x0001	For an animated stellar, the first frame will be shown after each subsequent frame.
0x0002	For an animated stellar, the next frame in the sequence will be picked at random. The same frame will not be picked twice in a row. Note that this can be combined with the previous flag and the Frame0Bias field to create interesting effects such as random planetary lightning or lights twinkling.
0x0010	Play this stellar's sound in a continuous loop.
0x0020	Stellar is always dominated (all your base are belong to us).
0x0040	Stellar starts the game destroyed.
0x0080	For an animated stellar, the stellar's graphic is animated after it's been destroyed and static when it is not destroyed. The normal behaviour is the opposite of this: static when destroyed and animated when not.
0x0100	Stellar is deadly - all ships that touch it are destroyed immediately.
0x0200	If the stellar has a weapon, it will only fire when provoked (i.e. only when the player is trying to dominate it).
0x0400	If the stellar has an outfit shop, it can buy any nonpermanent outfits the player owns, regardless of tech level.
0x1000	Stellar is a hypergate - if the player lands on it he will be given a choice of which other hypergate to travel to (see HyperLink1-8 below).
0x2000	Stellar is a wormhole - if the player lands on it he will be transported to some other random somewhere in the galaxy. If all of the wormhole's HyperLink1-8 fields set to -1, the player will end up at another random wormhole which also has no defined hyper links. If the wormhole has any hyper links defined, the player will end up at one of the wormholes on the other end.

AnimDelay	The time to wait between frames, in 30ths of a second.
Frame0Bias	If greater than 1, this is used as a multiplier to extend the display time of the first frame in the sequence. For example, a Frame0Bias of 3 would cause the first frame in the sequence would be held for three times longer than the rest of the frames.
HyperLink1-8	IDs of the spöb resources of up to eight other hypergates or wormholes to which this hypergate (or wormhole) is connected. Set to zero or -1 if unused. For a wormhole, setting every HyperLink field to -1 will cause the wormhole to randomly connect to any other random wormhole when the player goes through it.
OnDominate	Control bit set expression which is evaluated when the stellar is successfully dominated by the player.
OnRelease	Control bit set expression which is evaluated when the stellar is released from domination by the player/
Fee	The fee that is deducted from the player's credits when landing on this stellar. Set to zero if unused.
Gravity	The stellar's gravity - 0 for none, positive for stellars that pull, negative for stellars that push. Beware! This feature is mostly here for laughs. It severely confuses the AI, so it should only be used in empty systems where only the player can go.

Weapon	Stellars can have a single projectile or missile type weapon, with unlimited ammunition (don't put the ID of a beam or PD weapon here or bad things will happen). Stellars' weapons can be made to fire either only when provoked or any time an enemy ship is present.
0 or -1	No weapon.
128-383	Stellar has a weapon of this type.
Strength	The amount of combined mass and energy damage this stellar can take from planetary-type weapons before it is destroyed. Set this to 0 or -1 for an invincible stellar.
DeadType	Which stellar graphic to use when the stellar is destroyed.
-1	Don't display different graphic type when destroyed.
0-255	Display this stellar graphic when destroyed.
DeadTime	The amount of time a stellar remains destroyed before it regenerates itself. Set this to 0 for a stellar that regenerates at the end of every day, or -1 for a stellar that never regenerates on its own.
ExplodType	What kind of explosion to show when the stellar is destroyed.
-1	No explosion.
0-63	This type of explosion.
1000-1063	Explosion type 0-63, plus a random number of type-0 explosions around it.
OnDestroy	Control bit set expression which is evaluated when the stellar is destroyed.
OnRegen	Control bit set expression which is evaluated when the stellar automatically regenerates.

The sÿst resource

Syst resources store information on star systems, in which all combat, trading, and spaceflight take place. Each system can be linked to up to 16 other systems, and the player can make hyperspace jumps back and forth between them.

The first two fields in the syst resource tell Nova where on the map to place it:

xPos & **yPos**	The system's X and Y positions on the map.

The next fields store the hyperspace links to up to 16 other systems:

Con1-Con16	Link to another system.
-1	No link.
128-1127	ID of a system to link to.

The next fields store the stellar navigation defaults (F1-F4 for the first four) for the system. It is important to always set navigation defaults for stellar objects in your systems, because that's how Nova's AI routines and status display keep track of stellar objects; if you don't set a planet as a nav default, the AIs won't "see" it, it won't show up on the radar, and you can't select it.

NavDef (x16)	Navigation defaults (F1-F4).
-1	No nav default for this key.
128-548	ID number of the stellar object to set as a default.

The next fields tell Nova how many ships, and of what kind, to put in the system:

DudeTypes (x8)	Which type of dude to place (best not to set this to an out-of-range value).
128 to 639	ID number of the dude type to place.
-1	unused.
% **Prob** (x8)	Probability that a given ship placed is of each of the above types.
1-99	This percent probability.
AvgShips	The average number of AI ships in the system.
0	No ships, empty system.
1 and up	This number of ships, +/- 50%.

The next field tells Nova who controls the system:

Govt	Which government owns the system.
-1	Ignored (system is independent).
128-383	ID number of the controlling govt.

The next tells Nova which string, if any, to display as the message buoy's message when you enter a system:

Message	Which message buoy string to display.
-1	Ignored (no special message).
1 and up	Use this entry in STR# resource 1000 as the text of the message buoy.

The next two fields tell Nova what kinds of navigation hazards to put in the system:

Asteroids	How many asteroids to put in the system, from 0 to 16.
Interference	How thick the static in the system should be. 0 is no static, 100 is complete sensor blackout.

Want to make a 'pers' type ship always appear? Put its ID into one of the Person fields that appear at the end of the syst resource.

The Visibility field controls how and when to make the system visible or invisible. You can pull off some cool tricks with this field, including presenting the illusion that system has changed in some way by hiding the original system and replacing it with a copy that is identical except for the desired changes. The Visibility field is a control bit test expression – leave it blank if unused.

Note: Using the Visibility field to replace one system with another will work fine as long as they have the exact same coordinates (that's how Nova knows to update the hyper links, etc.) But, if you're going to have systems replacing each other in response to ncb changes, be sure that the Visibility fields of all the systems are mutually exclusive, or the resulting behaviour will be undefined and probably incorrect.

BkgndColor	The system's background colour, encoded the same as HTML colours. (RRGGBB) ...set to zero for pure black.
Murk	The murkiness of the system (0-100). Zero will cause everything to appear normally - 100 will cause the player to question their current glasses prescription. A value less than zero is equivalent to zero murk but also hides the starfield.

AstTypes Flag bits that determine what types of asteroids will appear in this system:

0x0001	Small metal	(röid ID 128)
0x0002	Medium metal	(röid ID 129)
0x0004	Large metal	(röid ID 130)
0x0008	Huge metal	(röid ID 131)
0x0010	Small ice	(röid ID 132)
0x0020	Medium ice	(röid ID 133)
etc.		

ReinfFleet	The ID of a fleet to use as this system's reinforcement fleet. If ships allied with the reinforcement fleet's government are under attack and the combat odds against them exceed the MaxOdds field of the reinforcement fleet's government, the reinforcement fleet will be called in. Set to 0 or -1 if unused.
ReinfTime	The delay between the time the reinforcement call goes out and the time the fleet appears. A value of 30 = one second.
ReinfIntrval	The interval, in days, that it takes for the reinforcement fleet to be regenerated. If you set this to 0, a reinforcement fleet will be available every day. If you set this higher than zero, it will take that number of days for the reinforcement fleet to be available again.

The wëap resource

The weap resource, surprisingly, stores info on Nova's weapons. The name of the weap resource is used as the weapon name in the weaponry section of the status display. The first two fields control the duration of different aspects of the weapon:

Reload	The number of frames it takes for one of this weapon to reload. 30 = 1 shot/sec. Smaller numbers yield faster reloads.
Count	The number of frames the weapon's shots travel for before they peter out. 30 = 1 second of life.

The next two fields, **MassDmg** and **EnergyDmg**, tell Nova how much damage to do when one of this weapon's shots hits something. Energy damage does damage to shields only, and mass damage does damage to armour only. Further, if a ship has shields, its armour can't be damaged until its shields are knocked down (unless the weapon is a shield-penetrating weapon).

The next two fields tell Nova how the weapon should behave in flight:

Guidance	The weapon's guidance mode.
-1	Unguided projectile.
0	Beam weapon (see below).
1	Homing weapon (see Seeker field below).
2	(unused).
3	Turreted beam.
4	Turreted, unguided projectile.
5	Freefall bomb (launched at 80% of the ship's current velocity, "weathervanes" into the "wind."
6	Freeflight rocket (launched straight ahead, accelerates to its maximum velocity).
7	Front-quadrant turret, (can fire +/-45¡ off the ship's nose) fires straight ahead if no target.
8	Rear-quadrant turret (can fire +/-45¡ off the ship's tail).
9	Point defence turret (fires automatically at incoming guided weapons and nearby ships).
10	Point defence beam (fires automatically at incoming guided weapons and nearby ships).
99	Carried ship (AmmoType is the ID of the ship class).
Speed	The weapon's speed (pixels per frame * 100).

The next field tells Nova how to handle the ammunition for this weapon, assuming it’s not a fighter bay:

AmmoType	What kind of ammo the weapon uses.
-1	Ignored (unlimited ammo).
0-255	Draws ammo from this type of weapon. (Usually, if your Hector Cannon was of ID 131, you'd set the AmmoType to 3 so it'd use Hector Birdseed Pellets or whatever. However, you could conceivably set it to use ammo from another weapon's supply by setting the AmmoType to something else.)
-999	Ship is destroyed when weapon is fired.
-1000 & below	Weapon uses abs(AmmoType+1000)/10 units of fuel per shot. (example: -1005 = 0.5 units per shot).

The next three fields tell Nova which graphic and sound to use for this weapon, and how to launch it:

Graphic	What graphic set to use for this weapon.
0-255	Use this graphic set (i.e. spin resources 3000-3255).

Inaccuracy	The weapon's inaccuracy as it leaves the ship (ignored for guidance-10 point defence beams).
0	Fires straight.
1 and up	Fires with up to this amount of inaccuracy (in degrees).

Sound	Which sound to play when the weapon fires.
-1	Silent but deadly.
0-63	Play this sound (snd ID 200-263).

The next four fields store info on how the weapon behaves when it hits something:

Impact	The magnitude of the impact when the shot hits something.
0	No impact.
1 and up	This amount of impact, which is inversely proportional to the ship's mass. (e.g. Missile = 30).

ExplodType	What kind of explosion to show when the weapon hits.
-1	No explosion.
0-63	This type of explosion.
1000-1063	Explosion type 0-63, plus a random number of type-0 explosions around it.

ProxRadius	The radius of the weapon's proximity fuse (useful for unguided missiles and bombs).
0	Weapon requires direct hit to do damage.
1 and up	This number of pixels of proximity radius.

BlastRadius	The radius of the weapon's blast effect.
0	No blast effect.
1 and up	This number of pixels of blast radius.

The next field contains some miscellaneous flag info:

Flags	
0x0001	Spin the weapon's graphic continuously (rate of frame advance is controlled by the BeamWidth field as detailed below).
0x0002	Weapon fired by second trigger.
0x0004	For cycling weapons, always start on the first frame of the animation.
0x0008	For guided weapons, don't fire at fast ships (ships with turn rate > 3).
0x0010	Weapon's sound is looped rather than played repeatedly.
0x0020	Weapon passes through shields (use sparingly!).
0x0040	Multiple weapons of this type fire simultaneously.
0x0080	Weapon can't be targeted by point defence systems (works only for homing weapons).
0x0100	Weapon's blast doesn't hurt the player.
0x0200	Weapon generates small smoke.
0x0400	Weapon generates big smoke.
0x0800	Weapon’s smoke trail is more persistent.
0x1000	Turreted weapon has a blind spot to the front.

`0x2000`	Turreted weapon has a blind spot to the sides.
`0x4000`	Turreted weapon has a blind spot to the rear.
`0x8000`	Shot detonates at the end of its lifespan (useful for flak-type weapons).

The next field contains flags that control how a guided weapon (Guidance = 1) behaves, as well as some misc flags for other weapon types:

Seeker

`0x0001`	Passes over asteroids.
`0x0002`	Decoyed by asteroids.
`0x0008`	Confused by sensor interference.
`0x0010`	Turns away if jammed.
`0x0020`	Can't fire if ship is ionized.
`0x4000`	Loses lock if target not directly ahead.
`0x8000`	May attack parent ship if jammed.

SmokeSet	Which cicn set to use for this weapon’s smoke trail, if any. 0 = cicn's 1000-1007, 1 = 1008-1015, etc. Note that the smoke icons themselves can be any size, but if you use ResEdit's cicn editor to make each icon only as large as it needs to be, game performance will likely improve.
Decay	How fast to decay each shot's power.
`-1 or 0`	Ignored.
`1 and up`	Remove one point of mass & energy damage every time this number of frames goes by (1 frame = 1/30 sec.).
Particles	Number of particles to generate per frame. Set to zero for no particles.
PartVel	Particle velocity, from 1 to about 200 or so. Experiment to find useful values.
PartLifeMin	Minimum life of a particle from this weapon, in frames.
PartLifeMax	Maximum particle life, in frames.
PartColor	Particle base colour, encoded the same as HTML colours. (RRGGBB).
BeamLength	The length of the beam created by this weapon, if applicable.
BeamWidth	Beam width (actually, radius) in pixels. A BeamWidth of 0 will have no centre beam, just corona glow.

Note: Lightning beams require a BeamWidth or 1 or greater.

Note: For sprite-based weapons that spin continuously, this field controls the time between frames, in 30ths of a second.

Falloff	The corona falloff. Higher numbers make the corona fall off faster. This value must be between 2 and 16. (since lightning beams have no corona, this is ignored for lightning beams)
BeamColor	The colour of the beam centre, encoded as a 24-bit RGB value (00RRGGBB).
CoronaColor	The colour of the beam corona, encoded as a 24-bit RGB value. Note that since the corona is translucent, it will appear approximately half as bright at

its maximum as what you specify in this field. (since lightning beams have no corona, this is ignored for lightning beams)

SubCount	The number of submunitions to create when the shot reaches the end of its life or detonates because something wanders into its proximity radius. Set to 0 or -1 if unused.
SubType	The resource ID of the weapon type to create as submunitions. Anything except beams and fighters is handled.
SubTheta	The angular error of the submunitions as they are launched, as expressed in degrees error from the carrier weapons' heading.
SubLimit	If you have defined a recursively-submunitioning weapon (i.e. one which splits into more copies of itself) this field will allow you to limit the number of recursive splits that happen. This field is ignored if the weapon is not recursively submunitioning.
ProxSafety	A time delay for the weapon's proximity fuse, in 30ths of a second. Set to zero for immediate arming on launch.

Flags2

0x0001	For cycling weapons, keep the graphic on the first frame until the weapon's ProxSafety count has expired.
0x0002	For cycling weapons, stop the graphic on the last frame.
0x0004	Proximity detonator ignores asteroids.
0x0008	Proximity detonator is triggered by ships other than the target (for guided weapons).
0x0010	Submunitions fire toward nearest valid target.
0x0020	Don't launch submunitions when the shot expires.
0x0040	Don't show weapon's ammo quantity on the status display.
0x0080	This weapon can only be fired when there is at least one ship of this ship's KeyCarried type aboard.
0x0100	AI ships won't use this weapon.
0x0200	This weapon uses the ship's weapon sprite, if applicable.
0x0400	Weapon is a planet-type weapon, and can only hit planet-type ships or destroyable stellars.
0x0800	Don't allow this weapon to be selected or displayed if it is out of ammo.
0x1000	Weapon can disable but not destroy.
0x2000	For beam weapons, display the beam underneath ships instead of on top of them.
0x4000	Weapon can be fired while cloaked.
0x8000	Weapon does x10 mass damage to asteroids.

Ionization	The amount of ionization energy to add to the ship that gets hit by this weapon. When a ship is ionized it becomes nearly immobilized until the ionization charge dissipates.
HitParticles	The number of particles to generate when a shot of this type hits a ship or asteroid. Set to zero if unused.
HitPartLife	The average life (in frames) of the hit-particles.
HitPartVel	The speed of the hit-particles; a value of 100 is one pixel per frame.

HitPartColor	The colour of the hit-particles, encoded as a 24-bit HTML colour (OORRGGBB).
ExitType	The type of weapon exit point to use for this weapon:
-1	Ignored - weapon will fire from the centre of the ship.
0	GunPosX/Y.
1	TurretPosX/Y.
2	GuidedPosX/Y.
3	BeamPosX/Y.
BurstCount	The number of shots this weapon can fire before having to endure a burst reload. Set to 0 or -1 if unused. For weapons that do not fire simultaneously, this value will be multiplied by how many of this weapon the firing ship has - for example, a weapon with a BurstCount of 4 on a ship with 3 of that weapon will be able to fire 12 times before entering the burst reload period. For weapons that fire simultaneously, this value is independent of how many of the weapon the ship has.
BurstReload	The reload time that is imposed when the weapon has fired >= BurstCount shots. Ignored if BurstCount is 0 or -1.
JamVuln1-4	The weapon's vulnerability to the four different types of jamming, from 0 to 100%. Ignored if the weapon is not a guided weapon.
Flags3	
0x0001	Weapon will only use ammo at the end of a burst cycle.
0x0002	Weapon's shots are translucent
0x0004	Firing ship can't fire another shot of this type until the previous one expires or hits something.
0x0010	Weapon fires from whatever weapon exit point is closest to the target.
0x0020	Weapon is exclusive - no other weapons on the ship can fire while this weapon is firing or reloading.
Durability	For guided weapons, this is how many point defence hits a shot from this weapon can take before it is destroyed. Set to 0 for weapons that are immediately destroyed by any PD hits. Ignored for non-guided projectile weapons or beams. (PD damage to guided weapons is calculated as 100% of mass damage plus 50% of energy damage).
GuidedTurn	For guided weapons, this is the turning speed of the weapon. Higher values yield more manoeuvrable missiles. Ignored for anything but guided weapons.
MaxAmmo	For ammo-using weapons, this is the maximum amount of ammo per each instance of this weapon. (so, if you have two of these weapons, the max amount of ammo for that weapon type would actually be twice MaxAmmo, and so on) Set to 0 or -1 if you want the ammo quantity to be constrained by the oütf resource's Max field instead.
Recoil	The amount of recoil force to apply to the firing ship when this weapon is used:
0 or -1	No recoil.
Positive values	Thrust the ship backwards
Negative values	Thrust the ship forwards (note that the resulting change to the ship's velocity is inversely proportional to its mass).
LiDensity	For a beam weapon, entering zero in this field will make it a normal, straight beam weapon. Entering a value greater than zero will make the

	beam a lightning beam, which has no real effect other than to make it look cool. The number you enter here is the number of zigzags the beam will make per 100 pixels. Higher numbers yield more convoluted beams. Note also that lightning beams can't have a beam corona, and so will only use the CoreColor and CoreWidth fields above.
LiAmplitude	The amplitude of each zigzag of a lightning beam, in pixels. Higher numbers yield more jagged-looking beams. Don't overdo this or you will have screen redraw problems. In fact, don't overuse lightning beams period, as they are much more processor-intensive to draw than normal beams!
IonizeColor	The colour that a ship hit by this weapon will appear after being sufficiently ionized (encoded the same as an HTML colour value). A value of 0 here will be interpreted as a default bluish colour. Using fairly bright colours here is probably the best, as low-intensity colours tend to look odd when used as the ionization colour.

Please note that if you've set the weapon to be a beam (Guidance of 0 or 3) the following fields have different functions:

Count	The number of frames the beam stays onscreen. Note also that if the beam has a positive Decay value entered (see below) the actual time the beam will exist onscreen will be Count + 16 - CoronaFalloff, so adjust the Reload value accordingly to ensure that the universe doesn't get filled up with overlapping beams from a single ships.
Impact	Functions normally, with one exception: if the impact is set to a negative value, the beam acts as a tractor beam. Smaller ships will be pulled towards the firing ship with a strength proportional to the Impact value, while a small ship firing a tractor beam at a larger ship (or asteroid) will "latch on" to it and be dragged along. Note that you cannot enter hyperspace if held by a tractor beam from a ship that's larger than you are. Note also that inertialess ships are not affected by tractor beams.
ProxRadius	Ignored.
BlastRadius	Ignored.
Decay	If Decay is greater than zero, the beam will "shrink" before it disappears from the screen. The actual time the beam spends on screen will be Count + 16 - CoronaFalloff in this case, so adjust your Reload value accordingly (otherwise you'll get too many beams onscreen at once).

Another Note: If you don't create an oütf resource for each weapon type, your ship's weapon load out will be corrupted when you land on a planet. (specifically, Nova will mistakenly remove any weapons for which you didn't create oütf resources) Also, having multiple outfit items that bestow the user with the same weapon type can cause weird things to happen.

Appendix I - Combat Ratings

Your combat rating is based on the number of kills you have made, which is the sum of the strengths of all the ships you have destroyed, times some internal multiplier for adjustment. The scale is as follows:

Kills:	Rating:
0	No Ability.
1	Little Ability.
100	Fair Ability.
200	Average Ability.
400	Good Ability.
800	Competent.
1600	Very Competent.
3200	Worthy of Note.
6400	Dangerous.
12,800	Deadly.
25,600	Frightening.

The text strings listed above are given only by way of illustration, since they can be changed by editing STR# 138.

Appendix II - Legal Status

Your legal status in a system is based on the crime tolerance of that system's government. (if the system is independent, it is based on the first government's [ID 128] crime tolerance) On this scale, enough "good" or "evil" points to equal the government's crime tolerance is given a value of 1:

Good Scale:	Legal Status:
0	Clean.
4	Citizen.
16	Good Citizen.
64	Upstanding Citizen.
256	Leading Citizen.
1024	Model Citizen.
4096	Virtuous Citizen.

Evil Scale:	Legal Status:
0	No record.
1	Minor Offender.
4	Offender.
16	Criminal.
64	Wanted Criminal.
256	Fugitive.
1024	Hunted Fugitive.
4096	Public Enemy.

The text strings listed above are given only by way of illustration, since they can be changed by editing STR# 134.

Appendix III - Patching STR# Resources

The STR# resource format used to store many of the strings in Nova may seem to be incompatible with the flexible nature of plug-in files. For example, a plug-in for a new ship would have to replace several of the built-in STR# resources to incorporate its new name into the game. The problem arises when you want to use two plug-ins that try to modify the same STR# resource.

The solution is not to change the STR# resources at all, but to use Nova's handy string patching functionality by updating only select strings in a STR# resource by providing Nova with a properly-numbered 'STR ' resource. For example, to change the first cargo type from food to something else, you'd simply create a 'STR ' resource with the ID 9000 and type in the name of your new commodity. A chart of 'STR ' resource numbers is provided below:

String Type:	Replacement 'STR ' ID range:
Message buoys	1000-2500
Hail quotes	5000-5511
Stellar types	7000-7063
Base prices of commodities	9300-9305
Commodity abbreviations for status display	9400-9405
Govt-specific greetings for comm dialog	10000-12559 (first 10 for govt -1, second 10 for govt 0, third 10 for govt 1, etc.)
përs-resource CommQuotes	15000-15511

Appendix IV - Log Files

Nova supports two different kinds of debug log output that are useful to plug-in developers:

1. If you create a file called "debuglog.txt" in your Nova folder, a large amount of debug info will be dumped into this file as the game runs. Further, enabling this output also turns on some additional error checking, which can be very handy for finding errors in plug-ins.

2. If you create a file called "pilotlog.txt" in your Nova folder, every time a pilot file is opened its contents will be written in human-readable form to the pilotlog.txt file. (works only after registering).