

PRIVATEER

ASCETIC SECTOR

QUEST MAKER'S MANUAL

February 8th, 2009

TABLE OF CONTENTS

Overview	2
Syntax	3
Reference Manual	
<i>OnLoad</i>	7
<i>Bartender</i>	7
<i>ClearNav</i>	8
<i>Mission</i>	8
<i>Ship</i>	9
<i>ShipDialogue</i>	12
<i>Character</i>	14
<i>CharacterDialogue</i>	17
<i>Trigger</i>	19

OVERVIEW

The Privateer: Ascii Sector quest scripting language allows players to create their own stories in the Ascii Sector universe. To create a quest script all you need is three things: a raw text editor such as Notepad, the Quest Maker found in the Privateer: Ascii Sector game, and this manual. You simply write your script into a .txt file using your editor of choice and then compile the script to a playable .qst file using the built-in Quest Maker in the game itself (access it from the game's main menu). You can then give it to other players to play by placing the .qst file in their 'quests' folder and then loading the quest by pressing 'Q' while landed in the game.

Every quest script is made up of three parts:

1) Quest information

- Title
- Author
- Required
- Achievement

2) Declaration and setting of variables (max. 10 in a script)

3) Quest scenes

- OnLoad
- Bartender
- ClearNav
- Missions (max. 5 in a scene)
- Ships (max. 30 in a scene)
- ShipDialogues (max. 30 in a scene)
- Characters (max. 10 in a scene)
- CharacterDialogues (max. 10 in a scene)
- Triggers (max. 25 in a scene)

The quest information in the beginning of the script is for displaying the quest name (Title) and author name (Author) in the quest menu in the game, as well as optionally making the quest dependent on first completing another quest (Required), and optionally specifying a short text that will be added to the Achievements section of the player's Quine5000 when the quest has been completed (Achievement).

After the quest information, any variables that are used in the script must be declared and given an initial value. Variables can be used for pretty much anything – from keeping track of how many pirates the player has killed, to whether or not he's talked to a certain individual, to how much grain he's bought on Basin...

Finally we get to the meat of the script: the scenes. Everything that makes up the story of the quest is described in separate scenes. Ships and characters are placed, missions are declared, triggers are set up, dialogue is written, and so on. Just like scenes in a movie or play, the scenes in a quest script can have different characters and different triggers that result in new things happening to drive the story forward.

Writing your first quest can seem pretty daunting, so it's a good idea to start small to get an idea of how everything works. Try writing a short script with a character and some dialogue, or a ship and some dialogue or a battle. Or try playing around with variables, triggers and timers by having the script change to another scene when the player has killed a number of pirates, talked to a specific character, or walked around on a specific base for a specific amount of time. You can also play through the included "Kilrathi Hunt" quest and then look at its script file (khunt.txt located in your 'quests' folder) to see a basic example of a working quest.

Should you have any questions or comments, please don't hesitate to make a post in the forums at www.asciisector.net/forums

SYNTAX

Below is an overview of the syntax of the Privateer: Ascii Sector quest scripting language. If this is the first time you lay eyes on this manual or a script, it can probably seem quite daunting, but just keep in mind that this shows *all* the available commands and functions. For your first quests, you will probably only use a fraction of these, so don't despair! To get detailed information about any of the functions seen below, turn to the Reference Manual part of this manual.

Everything written in `Courier` is the name of the script functions and everything in *italics* is the values that you specify. The function name and the subsequent value(s) are always separated by a colon (:). If something is in a parenthetical, it means that this information is optional.

In scenes, everything is divided into blocks. For example, to place a spaceship in a scene, you make a ship block by writing SHIP followed by all the information about the ship on separate lines, and then closing the block by writing /SHIP on the final line.

Title : *title of quest*
 Author : *name of quest author*
 Required : *title of required quest* : *author of required quest*
 Achievement : *achievement text*

variable name : *initial variable value*

SCENE *scene name*

```

ONLOAD
  MarkNavPoint : system name : nav point number
  RemoveMission : mission name
  RemoveNavPointMark : system name or All : nav point number
  Timer : timer number : Start
                                Stop
                                Set : value
  Variable : variable name : value
/ONLOAD

```

```
BARTENDER
    Exemption : base name
               message
/BARTENDER
```

CLEARNAV
 system name : *nav point number*
 /CLEARNAV

```
MISSION mission name
  Description : mission description
  Objective   : mission objective description  (: empty lines above)
  Expiration  : hours until mission expires
/MISSION
```

SHIP

Armor : 1 - 5
Attitude : -125 to 125
ECM : 0 - 3
Experience : *Rookie, Pro or Ace*
Faction : *Confed, Hunter, Kilrathi, Merchant, Militia, Pirate or Retro*
Gun : *Laser, Mass Driver, Meson Blaster, Neutron Gun, Particle Cannon, Ionic Pulse Cannon, Tachyon Cannon, Plasma Gun, Fusion Cannon or None*
GunBoost : 0 - 3
GunCooler : 0 - 3
MissileAmount : 0 - 20
MissileType : *Dumb, Heat, Forf or Torp*
Morale : *Timid, Confident or Fanatical*
Name : *ship name*
Nav : *nav point number*
Order : *Attack, Dock, DockNoEngage, Escort1-8, Jump, JumpNoEngage, Land, LandNoEngage, MoveTo, MoveToNoEngage or None*
OrderTarget : *target name or number*
Shields : 1 - 5
System : *system name*
TurretLeft : *Laser, Mass Driver, Meson Blaster, Neutron Gun, Particle Cannon, Ionic Pulse Cannon, Tachyon Cannon, Plasma Gun, Fusion Cannon or None*
TurretLeftBoost : 0 - 3
TurretLeftCooler : 0 - 3
TurretRear : *Laser, Mass Driver, Meson Blaster, Neutron Gun, Particle Cannon, Ionic Pulse Cannon, Tachyon Cannon, Plasma Gun, Fusion Cannon or None*
TurretRearBoost : 0 - 3
TurretRearCooler : 0 - 3
TurretRight : *Laser, Mass Driver, Meson Blaster, Neutron Gun, Particle Cannon, Ionic Pulse Cannon, Tachyon Cannon, Plasma Gun, Fusion Cannon or None*
TurretRightBoost : 0 - 3
TurretRightCooler : 0 - 3
Type : *Broadsword, Centurion, Demon, Dralhti, Drayman, Galaxy, Gladius, Gothri, Kamekh, Nexus, Orion, Paradigm, Pegasus, Stiletto, Talon, Tarsus, Ulysses*
Template : *template number*
X : 0 - 46000
Y : 0 - 46000

/SHIP

SHIPDIALOGUE *ship name*

STEP *step number*
ChangeScene : *scene name*
Goto : *step to go to*
QuestDone (: *ShowMessage*)
Ship : *ship dialogue*
Timer : *timer number : Start, Stop or Set : value*
Variable : *variable name : new value*
Player : *go to step : player reply*

/STEP

/SHIPDIALOGUE

CHARACTER

Color : 1 - 15
CriticalWounds : *amount of critical wounds*
EquipmentHead : *equipment name (: equipment condition)*
EquipmentLeftBelt : *equipment name (: equipment condition or ammunition amount)*
EquipmentLeftHand : *equipment name (: equipment condition or ammunition amount)*
EquipmentLeftShoulder : *equipment name (: equipment condition or ammunition amount)*
EquipmentRightBelt : *equipment name (: equipment condition or ammunition amount)*
EquipmentRightHand : *equipment name (: equipment condition or ammunition amount)*
EquipmentRightShoulder : *equipment name (: equipment condition or ammunition amount)*
EquipmentTorso : *equipment name (: equipment condition)*
FirstName : *character's first name*
Gender : *Male or Female*
HealthHead : 0 - 100
HealthLeftArm : 0 - 100
HealthLeftLeg : 0 - 100
HealthRightArm : 0 - 100
HealthRightLeg : 0 - 100
HealthTorso : 0 - 100
HealthTotal : 0 - 124
HomeName : *base or ship name*
HomeType : *Base or Ship*
LastName : *character's last name*
Movement : *Global, Local or None*
Order : *Attack, None, Protect or Talk*
OrderTarget : *target name*
Precision : 1 - 200
Speed : 1 - 200
Strength : 1 - 200
Template : *template number*
X : 1 - 88
Y : 1 - 84

/CHARACTER

CHARACTERDIALOGUE *character name*

Character : *character dialogue*
Player : *player dialogue*
Goto : *step to go to*
IF *Accept, Timer, Variable, Credits*
 if conditional is true, conversation continues here
ELSE
 if untrue, it continues here
/ELSE
Reply : *go to step : reply message*
STEP *step number*
 Character : *character dialogue*
 Reply : *go to step : reply message*
/STEP

/CHARACTERDIALOGUE

TRIGGER

Event : AreaEntered : *area (: base name)*
 CargoBought : *cargo type (: base name)*
 CargoInHold : *cargo type : amount*
 CargoSold : *cargo type (: base name)*

Reference Manual

ONLOAD

You can use an OnLoad block to have the script do a few things when the scene in which the block is placed is first loaded. You can add as many commands to the OnLoad block as you wish.

MarkNavPoint : *system name* : *nav point number*

This will cause the specified nav point in the specified system to show up as red on the nav computer (just like nav points that constitute a mission objective).

RemoveMission : *mission name*

This will remove the specified mission from the player's Quine and nav computer and free up its mission slot. Remember that you cannot have more than 5 missions in use at the same time – that's why it's important to remove them when they're no longer needed in the quest (also, they clutter the player's list of missions, since the player cannot remove quest missions from the Quine himself).

RemoveNavPointMark : *system name or All* : *nav point number*

This will cause the specified nav point in the specified system to no longer show up as red on the nav computer. If you want to remove the markings on all nav points, just specify All instead of a system name and nav point number.

Timer : *timer number* : *Start, Stop or Set* : *value*

This will Start, Stop or Set the timer with the specified number (1 - 5) to the specified value. When a timer has been started, it will be increased with 1 each game second. Stopping a timer will not reset it, so starting it again will make it continue from the value it had when it was stopped. When setting the timer value, you can specify the timer value as X, -X or +X. Timers are unaffected by scene changes.

Variable : *variable name* : *value*

This will set the variable with the specified name to the specified value. The value can be specified as X, -X or +X. You can only use the name of a variable you declared in the beginning of the quest script.

BARTENDER

When the player talks to the various bartenders in the game, you can have them tell the player a message that is relevant to the quest (for example what to do to get the quest underway). Using a Bartender block, you can specify the message for the bartender to tell the player, and you can specify an "exemption base", where the bartender won't tell the player this message (if, for example, the message tells the player to go to Loye Naval Base, you can tell the script not to have the bartender on Loye Naval Base say this to the player, as that would seem odd).

Exemption : *base name*

This is optional and will cause the bartender on the specified base not to tell the player the message.

message

Write the message itself inside the block. The message cannot be more than 10 lines long and each line cannot have more than 200 characters in it. When the bartender tells the player the message, any line breaks are ignored, so you don't have to worry about fitting the entire message on a single line in the script.

CLEARNAV

When the scene is first loaded, you can specify a number of nav points that shouldn't have random ships placed on them during this scene. You can specify as many nav points as you want, in as many different systems as you want. This is useful if you want to make sure that a quest ship isn't attacked and destroyed by a randomly generated enemy. Ships used for the player's active missions will still be generated, however.

system name : nav point number

Specify the system name and the nav point number of the nav point where you don't want randomly generated ships to be placed. You can have as many of these in the ClearNav block as you want.

MISSION *mission name*

When the player is given stuff to do during the quest, you'll probably want this to show up as missions in the player's Quine and nav computer. With a Mission block, you can create a mission with a description for the Quine and with objectives for the nav computer, just like any other missions in the game. When you write the mission block, the first thing to do is give the mission a name. This name will only be used when referencing the mission in the script – it won't ever show up in the game. Missions are unaffected by scene changes, so you can always reference a scene that has been created in a previous scene (unless you've since removed it, of course). You cannot have more than 5 missions in use at the same time – that's why it's important to remove them when they're no longer needed in the quest (also, they clutter the player's list of missions, since the player cannot remove quest missions from the Quine himself).

Description : *mission description*

The mission's description is what will show up in the player's Quine.

Objective : *mission objective description (: empty lines above)*

Each mission can have up to 9 objectives, which will be displayed in the nav computer, telling the player what to do to complete the mission. Objectives are automatically numbered in the order they are declared in the script. So, if you later need to reference a specific mission objective (for example in a trigger), you can get its number simply by counting the order in which it was declared here in the mission block. The optional parenthetical is if you want the mission objective to have blank lines above it when displayed in the nav computer (for example to visually separate it from the other objectives).

Expiration : *hours until mission expires*

If you want a mission to have an expiration date, you can specify how many hours the player has to complete it. The expiration date is automatically calculated by adding this amount of hours to the current game time when the scene with the mission is loaded.

SHIP

To create a ship in a scene, you make a ship block and inside this block specify the details of the ship, such as name, placement, weaponry, order and so on. A ship will be created when the player enters the system in which the ship is placed – either by jumping into the system or by launching from a base in the system. This means that the ship doesn't immediately appear when the scene is loaded – it just tells the game to place this ship in its system, when the player visits that system. If you want to immediately create a ship, you will have to use the CreateShip trigger result. Likewise, if you're in a system with a quest ship, even though another scene that doesn't contain this ship is loaded, the ship will remain in play until the player leaves the system and re-enters. If you want the same ship to appear in subsequent scenes, you don't have to retype all the information, you can just create a ship block with the ship's name – the game engine will remember all the information that that ship previously had. Please note, though, that if you have a scene in between the scene where the ship was first declared and the scene where you want to re-use the ship, and that in-between scene doesn't declare that ship, but another ship, this new ship will overwrite the original ship's information. So, if you want to make sure that the game remembers the original ship, always have a ship block with this ship's name in each scene to tell the game to keep this ship's information in memory.

Armor : *1 - 5*

This specifies the armor level of the ship. If you don't specify an armor level, the game will use level 1, which is the default amount of armor for this ship type. Armor level 2 is double the ship's default armor, 3 is triple, and so on.

Attitude : *-125 to 125*

This specifies the ship's starting attitude towards the player. If you don't specify an attitude, the game will use the overall attitude of the faction that the ship belongs to. -125 to -20 is hostile; -19 to 19 is neutral; 20 to 125 is friendly.

ECM : *0 - 3*

This specifies the ship's level of electronic counter measures against locking missiles. If you don't specify a level (or specify 0), the ship won't have any ECM.

Experience : *Rookie, Pro or Ace*

This specifies the experience of the ship (or its pilot), which affects how good the ship is at aiming at you, as well as making evasive maneuvers. If you don't specify the experience level, the game will set the default level of Pro.

Faction : *Confed, Hunter, Kilrathi, Merchant, Militia, Pirate or Retro*

This specifies the ship's faction, which affects the ship's attitude towards you and other ships. If the ship is of a type that more than one faction uses, the faction will also affect the default armaments of the ship. You must always specify a faction the first time you declare the ship.

Gun : *Laser, Mass Driver, Meson Blaster, Neutron Gun, Particle Cannon, Ionic Pulse Cannon, Tachyon Cannon, Plasma Gun, Fusion Cannon or None*

This specifies the type of forward facing gun on the ship. If the ship is a capital sized ship with more than one forward facing gun, this will cause all forward facing guns to be of this type. If you don't specify a gun type, the game will use the default gun for this ship type.

GunBoost : *0 - 3*

This will specify the level of gun boost for the forward facing gun, which affects the range of the gun. If you don't specify a level (or specify 0) the gun's range won't be boosted. If the ship is a capital sized ship with more than one forward facing gun, this will specify the gun boost level of all the forward facing guns.

GunCooler : *0 - 3*

This will specify the level of gun cooler for the forward facing gun, which affects the refire delay of the gun. The higher the gun cooler level, the lower the refire delay will be, and the faster the gun will shoot. If you don't specify a level (or specify 0) the gun won't be cooled. If the ship is a capital sized ship with more than one forward facing gun, this will specify the gun cooler level of all the forward facing guns.

MissileAmount : *0 - 20*

This specifies how many missiles (or torpedoes) the ship will carry. If it only has one missile launcher, it won't be able to carry more than 10 missiles, but if it is a capital sized ship with two missile launchers, it can carry up to 20 missiles. If you don't specify a missile amount, the ship will have the default amount of missiles for its type.

MissileType : *Dumb, Heat, Forf or Torp*

This will specify the type of missiles for all the missiles the ship is carrying. If you don't specify a missile type, the ship will carry the default type of missiles for its type.

Morale : *Timid, Confident or Fanatical*

This specifies the morale level of the ship (or its pilot), which affects how probable it is for the ship to try to escape if its shield or armor level becomes critical. A Timid ship will be more prone to run off, while a Fanatical ship will never run off. A Fanatical ship may also try to kamikaze you. If you don't specify the morale level, the game will set the ship to the default level of Timid.

Name : *ship name*

This will specify a unique name for the ship. If you don't specify a name, it will just be called the name of the faction, like other non-unique ships in the game. Without a unique name, you won't be able to use the ship for trigger events and results, however, and you won't be able to re-declare it in other scenes by simply referring to it by name.

Nav : *nav point number*

This will specify at which nav point in the specified system the ship is to be placed. If you don't want to place it at a nav point, you can use the X and Y values to place it at an exact spot in a system.

Order : *Attack, Dock, DockNoEngage, Escort1-8, Jump, JumpNoEngage, Land, LandNoEngage, MoveTo, MoveToNoEngage or None*

This will specify the ship's initial order. If you don't specify an order, the ship will just fly around casually. Attack will make the ship attack its current target, Dock will make it dock with the target, DockNoEngage will make it not engage enemies while docking, Escort1-8 will make it escort the target at the specified position (1-8, with 1 being at the 12 o'clock position and moving clockwise) and attack any ships attacking the target; Jump will make the ship move to the jump point at the nav point specified as its target and jump out; JumpNoEngage will make it not engage enemies while moving to the jump point and jumping out; Land will make it move to the base at the nav point specified as its target and land; LandNoEngage will make it not engage enemies while moving to the base and landing; MoveTo will make it move to the nav point specified as its target and fly around casually when reaching it; MoveToNoEngage will make it not engage enemies while moving to the nav point; and None will make the ship fly around casually at its current location (engaging any enemies that may be there), similar to not specifying an order at all.

OrderTarget : *target name or number*

This will specify the target for the ship's initial order. It can either be the number of a nav point or the name of a ship (or Player if the target should be the player's ship). If the target is another quest ship, make sure that the target ship has already been declared in the script. Otherwise the game won't be able to select it as this ship's target.

Shields : *1 - 5*

This specifies the shield level of the ship. If you don't specify a shield level, the game will use level 1, which is the default amount of shield for this ship type. Shield level 2 is double the ship's default shield, 3 is triple, and so on.

System : *system name*

This specifies the system in which to place the ship. You must always specify a system the first time you declare the ship otherwise the game won't know where to place it (unless you also specify a template number and place the ship with the CreateShip trigger result).

**TurretLeft, TurretLeftBoost, TurretLeftCooler, TurretRear
TurretRearBoost, TurretRearCooler, TurretRight, TurretRightBoost
and TurretRightCooler**

This specifies the type of gun on the ship's turret, the gun booster or the gun cooler. Declare it as you would the forward facing gun. The small fighter sized ships can only have a rear turret, whereas the capital sized ships can have rear, left and right turrets.

Template : *template number*

If you specify a template number for a ship, it won't be placed automatically when you enter the ship's system. Instead, you can create the ship at any time just outside the player's radar by using the CreateShip trigger result. In this way, you can create as many ships with this ship's settings as you want.

X : 0 - 46000

Y : 0 - 46000

Instead of specifying a nav point for the ship to be placed at, you can specify the exact X and Y coordinates in the system to place the ship at. The X and Y coordinates can be anywhere between 0 and 46000, but only the range of 21050 to 25250 is within range of the nav computer. The upper left corner of a system is (0,0), while the lower right is (46000,46000). If a ship with a specified X and Y position is created with the CreateShip trigger result, it won't be placed just outside the player's radar, but at the (X,Y) position. To get the coordinates of your ship's current position hold down SHIFT+'X'.

SHIPDIALOGUE *ship name*

Dialogues in space are rather simple. When writing the ShipDialogue block, you first specify the name of the ship that this dialogue relates to. A ShipDialogue block is divided into separate steps in which the NPC ship sends some messages to the player and then the player will be able to select a reply. Each reply will continue the dialogue in another step, wherein the NPC ship has some new messages to send to the player, and after which the player will have new reply options. You can, however, also jump to a different step in the dialogue without the player having to select a reply by using the Goto function. You can also have QuestDone, ChangeScene, Variable and Timer calls within a ship dialogue, but these function calls must always come before any player replies or they won't work. Also, for a function call to work within a ship dialogue, the step that the call is placed within must have at least one line of dialogue, or the entire step is ignored (including the function call).

STEP *step number*

This will create a new step block with the specified number within the ShipDialogue block. If this is the first step block of the entire dialogue, how the conversation begins depends on whether or not the first line of dialogue is to be spoken by the NPC ship or selected by the player from the list of replies. If the first line of dialogue belongs to the NPC, then the conversation will automatically be initiated by the NPC ship when it is within the player's view. Otherwise the conversation won't begin until the player sends a message to the NPC ship.

ChangeScene : *scene name*

You can have a ChangeScene call in the middle of a ShipDialogue if it is placed before any player replies. This will immediately load the specified scene, clearing everything associated with the current scene. The dialogue will immediately end as the new scene is loaded.

QuestDone (: *ShowMessage*)

You can have a QuestDone call in the middle of a ShipDialogue if it is placed before any player replies. This will finish the quest and write the achievement (if any) to the player's Quine5000. It will also immediately end the dialogue, as all information associated with the quest is cleared. Optionally, you can have the game show a message that tells the player that the quest has been completed by adding ShowMessage.

Ship : *ship dialogue*

This will specify a line of dialogue for the NPC ship to send to the player. Remember that each line should be short enough to fit on the bottom of the screen. You can have as many lines as you want. If you want the NPC to address the player by name, you can use +FirstName+ and +LastName+ which will automatically be replaced with the player's first and last name when displaying the dialogue.

Timer : *timer number* : *Start, Stop or Set* : *value*

You can have a Timer call in the middle of a ShipDialogue if it is placed before any player replies. This will Start, Stop or Set the timer with the specified number (1 - 5) to the specified value. When a timer has been started, it will be increased with 1 each game second. Stopping a timer will not reset it, so starting it again will make it continue from the value it had when it was stopped. When setting the timer value, you can specify the timer value as X, -X or +X. Timers are unaffected by scene changes.

Variable : *variable name* : *value*

You can have a Variable call in the middle of a ShipDialogue if it is placed before any player replies. This will set the variable with the specified name to the specified value. The value can be specified as X, -X or +X. You can only use the name of a variable you declared in the beginning of the quest script.

Goto : *step to go to*

This will cause the dialogue to immediately jump to the step with the specified number. This is useful if you want the dialogue to continue in another step without the player having to choose a reply in order to jump to another step.

Player : *go to step* : *player reply*

This will create a reply option for the player with the specified text that will continue the dialogue in the specified step. The reply text should be short enough to fit in the reply box.

SHIPDIALOGUE EXAMPLE:

```
SHIPDIALOGUE Millennium Falcon
STEP 1
    Ship : Hailing ship... Is +FirstName+ +LastName+ on board?
    Player : 2 : Yes, that's me.
    Player : 4 : You've got the wrong ship.
/STEP
STEP 2
    Ship : Excellent. My name's Han Solo. I need your help.
    Player : 3 : I'm listening.
    Player : 3 : What do you need my help for?
    Player : 4 : I've got enough on my hands already.
/STEP
STEP 3
    Ship : It's not safe to talk here. Meet me on Basin.
    ChangeScene : MeetHanSoloOnBasin
/STEP
STEP 4
    Ship : Sorry to have disturbed you then...
/STEP
/SHIPDIALOGUE
```

CHARACTER

To create a character in a scene, you make a character block and inside this block specify the details of the character, such as name, color, placement, health and so on. A character will be created when the player lands at the character's home, which can either be a base or a ship (currently, characters can only be placed on bases, as you cannot yet board other ships in the game). This means that the character doesn't immediately appear when the scene is loaded – it just tells the game to place this character on its home base or ship, when the player lands there or boards it. If you want to immediately create a character, you will have to use the CreateCharacter trigger result. Likewise, if you're on a base or ship with a quest character, even though another scene that doesn't contain this character is loaded, the character will remain in play until the player leaves the base or ship. If you want the same character to appear in subsequent scenes, you don't have to retype all the information, you can just create a character block with the character's name – the game engine will remember all the information that that character previously had. Please note, though, that if you have a scene in between the scene where the character was first declared and the scene where you want to re-use the character, and that in-between scene doesn't declare that character, but instead declares another character, this new character will overwrite the original character's information. So, if you want to make sure that the game remembers the original character's information, always have a character block with this character's name in each scene to tell the game to keep this character's information in memory.

Color : 1 - 15

This will specify the color of the character (the color of the character's dot and dialogue text). You must always specify a color the first time you declare the character. From 1 to 15 the colors are: Navy, Green, Teal, Maroon, Purple, Brown, Silver, Gray, Blue, Lime, Aqua, Red, Fuchsia, Yellow, White.

CriticalWounds : *amount of critical wounds*

This will specify the character's initial number of critical wounds. Each critical wound will drain the character's health, so the more critical wounds, the faster the character will bleed to death.

**EquipmentHead, EquipmentLeftBelt, EquipmentLeftHand,
EquipmentLeftShoulder, EquipmentRightBelt, EquipmentRightHand,
EquipmentRightShoulder and EquipmentTorso** : *equipment name (: value)*

You can use this to give the character equipment. The equipment name is the name of one of the pieces of equipment that exist per default in the game universe – if you want the character to have a quest item you will have to use the Giveltem trigger result. If you're placing equipment on the character's head, it can only be a helmet – if on the torso, only a vest. For helmets and vests, the value you can optionally specify is the condition of the vest or helmet (from 1-100). If the equipment uses ammunition, you can instead specify the amount of ammunition in it (the amount cannot exceed the equipment's default maximum amount).

FirstName : *character's first name*

This will specify the character's first name. You must always specify a first name. When referencing the character in the script, you will be referencing it by its full name (first name + an empty space + last name). If the character doesn't have a last name, its first name will be its full name.

Gender : *Male or Female*

This will specify the character's gender. If you don't specify a gender, the character will be male. A character's gender doesn't yet have any effect on gameplay.

**HealthHead, HealthLeftArm, HealthLeftLeg, HealthRightArm,
HealthRightLeg, HealthTorso and HealthTotal** : *value*

This will specify the condition of the character's body parts and overall health. The overall health must be from 1 to 124, while the health of the various body parts can be from 0 to 100 (with 0 meaning that the body part is destroyed – if the head or torso is destroyed, the character is dead). If you do not specify the character's health, it will be at maximum level.

HomeName : *base or ship name*

Specify the name of the character's home. If it is a base, specify the name of the base (for example Basin, Loye Naval Base or New Beijing) – if a ship, the name of the ship (currently you can only place a character on a base, as the boarding of other ships hasn't yet been implemented).

HomeType : *Base or Ship*

This specifies whether the character is to be placed on a Base or on a Ship (currently you can only place a character on a base, as the boarding of other ships hasn't yet been implemented). You must always specify the home type the first time you declare the character.

LastName : *character's last name*

Specify the character's last name. When referencing the character in the script, you will be referencing it by its full name (first name + an empty space + last name). If you don't specify a last name, the character's first name will be its full name.

Movement : *Global, Local or None*

Specify how much the character should be able to move around. If Global, the character can walk anywhere on its base or ship; if Local, the character will stay inside its current area; if None, the character won't move at all (use this if you want it to seem as if the character is sitting on a chair).

Order : *Attack, None, Protect or Talk*

This will specify the character's order. Attack will make the character attack its current target; None will make the character just walk around casually; Protect will make the character stay close to the specified target and attack any characters that attack this target; and Talk will make the character walk up to the target and automatically engage in the quest dialogue associated with the character (the target for the Talk order must always be Player, as two NPCs can't talk to each other).

OrderTarget : *target name*

This will specify the target for the character's order. If the character's order is Talk, the target must always be Player, as two NPCs can't talk to each other.

Precision : *1 - 200*

The character's precision dictates how good the character is at using ranged weapons (for example shooting guns). The higher, the better. The average value is 50.

Speed : *1 - 200*

The character's speed dictates how fast the character will move around. During combat a character with a high speed value will also be able to get more kicks/hits/shots off against its opponent, and the opponent will have a harder time hitting it (you will miss more melee attack against an opponent with a high speed). The average value is 50.

Strength : *1 - 200*

The character's strength dictates how much damage it can do in a melee attack. Also, when melee attacking a character with a high strength value, more attacks will be blocked. The average value is 50.

Template : *template number*

If you specify a template number for a character, it won't be placed automatically when you land at the character's base or board its ship. Instead, you can create the character at any time by using the CreateCharacter trigger result. If you specify an X and Y position for the character, it will be placed there – otherwise it will be placed at a random spot outside the player's view. In this way, you can create as many characters with this character's settings as you want.

X : 1 - 88

Y : 1 - 84

This will specify the coordinates at which to place the character on its base or ship. You must specify an X and Y value if the character is supposed to be placed automatically when you visit its base or ship, but if you give it a template number and plan to place it with the CreateCharacter trigger result, you don't have to specify an X and Y value. To get the coordinates for placing a character, you can use the 'L'ook feature while in the game and then hold down SHIFT+'X' to get the coordinates of the cursor's position.

CHARACTERDIALOGUE *character name*

CharacterDialogues are somewhat different than ShipDialogues, as the player's character can say lines of dialogue without the player having to choose these lines. When writing the CharacterDialogue block, you first specify the name of the character that this dialogue relates to. You can then just write what you want the NPC to say by using the Character function, and what the player's character should say by using the Player function. However, to create branching dialogue and allow the player to have input into what should be said, you can use a variety of different methods. You can use If functions that will check whether or not the conditions have been met – if not, the game will ignore the dialogue inside the If block. You can optionally choose to add an Else block to the If block, so that the game will continue the dialogue inside the Else block only if the condition *isn't* met. You can use four different conditions that the If block should meet: Accept, Credits, Timer and Variable. That's one way to have branching dialogue. The other is to use Step blocks, just like in ShipDialogues. And just like in ShipDialogues, you can also have QuestDone, ChangeScene, Variable and Timer calls within a character dialogue. While these calls must always come before any player replies in a ShipDialogue block, they can be placed anywhere in a CharacterDialogue block.

Character : *character dialogue*

This will specify a line of dialogue for the NPC to say. If you can't fit all the dialogue on one line, just split it into multiple **Character** : *character dialogue* lines, as the game will automatically combine all these lines into one big piece of dialogue for the NPC to say. If you want the NPC to address the player by name, you can use +FirstName+ and +LastName+ which will automatically be replaced with the player's first and last name when displaying the dialogue.

Player : *player dialogue*

This will specify a line of dialogue for the player's character to automatically say. If you can't fit all the dialogue on one line, just split it into multiple **Player** : *character dialogue* lines, as the game will automatically combine all these lines into one big piece of dialogue for the player's character to say.

ChangeScene : *scene name*

You can have a ChangeScene call anywhere in a CharacterDialogue block. This will immediately load the specified scene, clearing everything associated with the current scene. The dialogue will immediately end as the new scene is loaded.

QuestDone (: *ShowMessage*)

You can have a QuestDone call anywhere in a CharacterDialogue block. This will finish the quest and write the achievement (if any) to the player's Quine5000. It will also immediately end the dialogue, as all

information associated with the quest is cleared. Optionally, you can have the game show a message that tells the player that the quest has been completed by adding ShowMessage.

Timer : *timer number : Start, Stop or Set : value*

You can have a Timer call anywhere in a CharacterDialogue block. This will Start, Stop or Set the timer with the specified number (1 - 5) to the specified value. When a timer has been started, it will be increased with 1 each game second. Stopping a timer will not reset it, so starting it again will make it continue from the value it had when it was stopped. When setting the timer value, you can specify the timer value as X, -X or +X. Timers are unaffected by scene changes.

Variable : *variable name : value*

You can have a Variable call anywhere in a CharacterDialogue block. This will set the variable with the specified name to the specified value. The value can be specified as X, -X or +X. You can only use the name of a variable you declared in the beginning of the quest script.

IF *Accept, Timer, variable name or Credits*

You can create an If block by first writing IF followed by the conditional on the same line, then followed by everything that should happen if the condition is true on separate lines. You can then either close the If block with /IF or you can continue into an Else block with ELSE. If the condition is false, the stuff in the Else block will happen instead. Close the Else block with /ELSE. To have the player choose Accept or Reject as the condition, just write IF Accept. If you want the condition to be the value of a timer (for example if timer number 2 is larger than 500), then write IF Timer 2 > 500. If you want the condition to be the value of a variable (for example if the variable called KilrathiKilled is larger than 5), then write IF KilrathiKilled > 5. And finally, if you want the condition to be the amount of credits the player has, simply write IF Credits > 100000.

Reply : *go to step : reply message*

This will create a reply option for the player with the specified text that will continue the dialogue in the specified step. You can have a maximum of 5 replies for the player to choose from.

Goto : *step to go to*

This will cause the dialogue to immediately jump to the step with the specified number. This is useful if you want the dialogue to continue in another step without the player having to choose a reply in order to jump to another step.

STEP *step number*

This will create a new step block with the specified number within the CharacterDialogue block.

CHARACTERDIALOGUE EXAMPLE:

```
CHARACTERDIALOGUE Han Solo
  Character : +FirstName+, right? Glad you came!
  Player : So, what is it you need my help with?
  Character : I seem to have lost my Wookie. He's tall and
  Character : brown and answers to the name 'Chewie'.
  Reply : 1 : Dude, I'm so not helping you with that.
  Reply : 2 : What the hell's a Wookie?
  Reply : 3 : And you want me to find him for you?
STEP 1
  Character : Oh, well I'm sorry to hear that,
  ChangeScene : RefusedToHelp
/STEP
STEP 2
  Character : It's pretty much a walking carpet.
  Character : But he helps me out.
  Goto : 3
/STEP
STEP 3
  Character : If you could find him, I'd be very grateful.
  Player : How grateful?
  Character : I'll give you some very important information
  Character : if you find my Wookie... and if you toss in
  Character : 5000 credits on top.
  IF Credits < 5000
    Player : I don't have 5000 credits!
    Character : Then come back when you do...
  ELSE
    Player : 5000 credits is a lot, you know.
    Character : I know. Do we have a deal or not?
    IF Accept
      Player : Yeah, we have a deal.
      Variable : Pay5000Credits : 1
      ChangeScene : FindChewie
    ELSE
      Player : Let me think about it...
      ChangeScene : NotYetAcceptedHanSoloMission
    /ELSE
  /ELSE
/STEP
/CHARACTERDIALOGUE
```

TRIGGER

Triggers are the core of any quest script. They consist of two basic elements: an Event and a Result. The event will cause the trigger to activate and have the specified result. Per default, triggers can only activate once, but you can also add a Repeats element and specify the number of times you want the trigger to be able to activate. Each trigger cannot have more than 5 events and 5 results. Only one of the events can activate the trigger – the others (if any) must be conditionals that check if certain requirements are met to activate the trigger (for example, if a variable has a certain value or a mission objective has been completed).

Event : AreaEntered : *area* (: *base name*)

Will activate the trigger when the player enters into the type of area specified. Valid areas are: Bar, Exchange, Hotel, Main, MercenariesGuild, MerchantsGuild and Shipyard. Optionally, you can specify a base name to only have the trigger activate when the player enters the specified area at the specified base.

Event : CargoBought : *cargo type* (: *base name*)

Will activate the trigger when the player buys a unit of the commodity specified as cargo type. Valid cargo types are the names of the 40 different commodities traded in the game. Optionally, you can specify a base name to only have the trigger activate when the player buys the commodity at the specified base.

Event : CargoInHold : *cargo type* : *amount*

Will activate the trigger when the player reaches the specified amount of the specified cargo in the cargo hold. Valid cargo types are the names of the 40 different commodities traded in the game. The amount to activate the trigger at can be specified as X, <X or >X. This event can also be used as a conditional along with another event to only activate the trigger when the other event happens *and* the player has the specified amount of the specified cargo.

Event : CargoSold : *cargo type* (: *base name*)

Will activate the trigger when the player sells a unit of the commodity specified as cargo type. Valid cargo types are the names of the 40 different commodities traded in the game. Optionally, you can specify a base name to only have the trigger activate when the player sells the commodity at the specified base.

Event : CharacterBlinded (: *character name*)

Will activate the trigger when a character is completely blinded. Optionally, you can have the trigger only activate when a character with the specified name is blinded. If you want that character to be the player, specify the character name as Player.

Event : CharacterHitByPlayer (: *character name*)

Will activate the trigger when a character is hit by the player in a melee attack. Optionally, you can have the trigger only activate when a character with the specified name is hit.

Event : CharacterKilled (: *character name*)

Will activate the trigger when a character is killed. Optionally, you can have the trigger only activate when a character with the specified name is killed.

Event : CharacterKilledByPlayer (: *character name*)

Will activate the trigger when a character is killed by the player. Optionally, you can have the trigger only activate when a character with the specified name is killed.

Event : CharacterOrderCompleted : *character name*

Will activate the trigger when the character with the specified name completes his/her current order. The order doesn't necessarily have to be one specified by the quest script.

Event : CharacterShotByPlayer (: *character name*)

Will activate the trigger when a character is shot by the player. Optionally, you can have the trigger only activate when a character with the specified name is shot by the player.

Event : CharacterStunned (: *character name*)

Will activate the trigger when a character is completely stunned. Optionally, you can have the trigger only activate when a character with the specified name is stunned. If you want that character to be the player, specify the character name as Player.

Event : Credits : *amount*

Will activate the trigger when the player reaches the specified amount of credits. The amount to activate the trigger at can be specified as X, <X or >X. This event can also be used as a conditional along with another event to only activate the trigger when the other event happens *and* the player has the specified amount of credits.

Event : ItemAcquired : *item name*

Will activate the trigger when the player gets the item with the specified name into his inventory. The item doesn't necessarily have to be one specified by the quest script.

Event : ItemLost : *item name*

Will activate the trigger when the item with the specified name is removed from the player's inventory – either voluntarily or involuntarily. The item doesn't necessarily have to be one specified by the quest script.

Event : Jumped (: *To or From* : *system name*)

Will activate the trigger when the player does a jump through a jump point. Optionally, you can have the trigger only activate when the player jumps To or From the specified system.

Event : Landed (: *base name*)

Will activate the trigger when the player lands at a base. Optionally, you can have the trigger only activate when the player lands at the specified base.

Event : Launched (: *base name*)

Will activate the trigger when the player launches from a base. Optionally, you can have the trigger only activate when the player launches from the specified base.

Event : Mission : *mission name* : *Completed, Expired or Failed* (: *objective number*)

Will activate the trigger when the specified mission is Completed, Expired or Failed. Optionally, you can have the trigger activate when a specific objective of the mission is Completed, Expired or Failed by specifying the number of the objective. This event can also be used as a conditional along with another event to only activate the trigger when the other event happens *and* the specified mission or mission objective has been Completed, Expired or Failed.

Event : NavVisited : *system name* : *nav point number*

Will activate the trigger when the player visits the specified nav point in the specified system.

Event : ShipDestroyed (: *ship name, ship type or faction name*)

Will activate the trigger when a ship is destroyed. Optionally, you can have the trigger only activate when the ship with the specified name, of the specified type or of the specified faction is destroyed. Valid ship types are the names of the various ship models in the game and valid faction names are: Confed, Hunter, Kilrathi, Merchant, Militia, Pirate and Retro.

Event : ShipDestroyedByPlayer (: *ship name, ship type or faction name*)

Will activate the trigger when a ship is destroyed by the player. Optionally, you can have the trigger only activate when the ship with the specified name, of the specified type or of the specified faction is destroyed by the player. Valid ship types are the names of the various ship models in the game and valid faction names are: Confed, Hunter, Kilrathi, Merchant, Militia, Pirate and Retro.

Event : ShipHitByPlayer (: *ship name, ship type or faction name*)

Will activate the trigger when a ship is hit by the player's gunfire or missile/torpedo. Optionally, you can have the trigger only activate when a ship with the specified name, of the specified type or of the specified faction is hit by the player's gunfire or missile/torpedo. Valid ship types are the names of the various ship models in the game and valid faction names are: Confed, Hunter, Kilrathi, Merchant, Militia, Pirate and Retro.

Event : ShipIdentified (: *ship name, ship type or faction name*)

Will activate the trigger when a ship is identified by the player (i.e. the ship's name, type or faction is displayed on the player's scanner). Optionally, you can have the trigger only activate when a ship with the specified name, of the specified type or of the specified faction is identified by the player. Valid ship types are the names of the various ship models in the game and valid faction names are: Confed, Hunter, Kilrathi, Merchant, Militia, Pirate and Retro.

Event : ShipOrderCompleted : *ship name*

Will activate the trigger when the ship with the specified name completes its current order. The order doesn't necessarily have to be one specified by the quest script.

Event : Timer : *timer number* : *value*

Will activate the trigger when the timer of the specified number reaches the specified value. The value to activate the trigger at can be specified as X, <X or >X. This event can also be used as a conditional along with another event to only activate the trigger when the other event happens *and* the timer has the specified value.

Event : Variable : *variable name* : *value*

Will activate the trigger when the specified variable reaches the specified value. The value to activate the trigger at can be specified as X, <X or >X. This event can also be used as a conditional along with another event to only activate the trigger when the other event happens *and* the variable has the specified value.

Result : ChangeCharacterOrder : *character name* : *order*

When the trigger is activated, this will change the specified character's current order. Valid orders are: Attack, None, Protect, Talk. Attack will make the character attack its current target (which can be changed with ChangeCharacterOrderTarget); None will make the character just walk around casually; Protect will make the character stay close to the specified target and attack any characters that attack this target; and Talk will make the character walk up to the target and automatically engage in the quest dialogue associated with the character (the target for the Talk order must always be Player, as two NPCs can't talk to each other).

Result : ChangeCharacterOrderTarget : *character name* : *target character name*

When the trigger is activated, this will change the target for the specified character's order. If the character's order is Talk, the target must always be Player, as two NPCs can't talk to each other.

Result : ChangeScene : *scene name*

When the trigger is activated, this will immediately load the specified scene, clearing everything associated with the current scene. This means that if you for example activate the trigger with this result during a dialogue, the dialogue will immediately end as the new scene is loaded.

Result : ChangeShipAttitude : *ship name* : *attitude*

When the trigger is activated, this will change the attitude of the ship with the specified name towards the player's ship. Valid attitudes are: -125 to -20 (hostile), -19 to 19 (neutral) and 20 to 125 (friendly).

Result : ChangeShipOrder : *ship name* : *order*

When the trigger is activated, this will change the current order of the ship with the specified name. Valid orders are: Attack, Dock, DockNoEngage, Escort1-8, Jump, JumpNoEngage, Land, LandNoEngage, MoveTo, MoveToNoEngage and None. Attack will make the ship attack its current target (which can be changed with ChangeShipOrderTarget), Dock will make it dock with the target, DockNoEngage will make it not engage enemies while docking, Escort1-8 will make it escort the target at the specified position (1-8, with 1 being at the 12 o'clock position and moving clockwise) and attack any ships attacking the target; Jump will make the ship move to the jump point at the nav point specified as its target and jump out; JumpNoEngage will make it not engage enemies while moving to the jump point and jumping out; Land will make it move to the base at the nav point specified as its target and land; LandNoEngage will make it not engage enemies

while moving to the base and landing; MoveTo will make it move to the nav point specified as its target and fly around casually when reaching it; MoveToNoEngage will make it not engage enemies while moving to the nav point; and None will make the ship fly around casually at its current location (engaging any enemies that may be there). When calling both ChangeShipOrder and ChangeShipOrderTarget on the same ship *always* call ChangeShipOrder first, as the value passed with ChangeShipOrderTarget is processed based on the ships current order.

Result : ChangeShipOrderTarget : *ship name : target name or number*

When the trigger is activated, this will change the target for the specified ship's order. If the target is another ship, specify the target ship's name. If it is a nav point, specify the number of the nav point. When calling both ChangeShipOrder and ChangeShipOrderTarget on the same ship *always* call ChangeShipOrder first, as the value passed with ChangeShipOrderTarget is processed based on the ship's current order.

Result : CreateCharacter : *template number*

When the trigger is activated, this will create the character with the specified template number. The character will be created at the X and Y position specified in the character's template. If the X and Y positions are both 0, the character will be created at a random spot outside the player's current view.

Result : CreateItem : *item number : item name*

When the trigger is activated, this will create an item with the specified name in the item slot with the specified number (1-4). If you choose an item slot with an already existing item, it will be overwritten with this item.

Result : CreateShip : *template number*

When the trigger is activated, this will create a ship equal to the ship with the specified template number. The ship will be created at a random spot just outside of the player's radar range.

Result : GiveItem : *item name : character name*

When the trigger is activated, this will move the specified item from wherever it is (if it is already in play) and place it in the inventory of the specified character. If the character's inventory is full, the item in the character's right hand will be dropped to the ground and the item placed here. If you want the item given to the player, specify the character name as Player.

Result : MarkNavPoint : *system name : nav point number*

When the trigger is activated, this will cause the specified nav point in the specified system to show up as red on the nav computer (just like nav points that constitute a mission objective).

Result : Mission : *mission name : Completed, Expired or Failed (: objective number)*

When the trigger is activated, this will set the specified mission as Completed, Expired or Failed. Optionally, you can set one of the mission's objectives as Completed, Expired or Failed by specifying the objective number. When all objectives of a mission have been completed, the mission itself is automatically set to Completed. If an objective is Expired or Failed, so is the mission.

Result : PlaceItemOnBase : *item name* : *base name* : *x position* : *y position*

When the trigger is activated, this will place the item of the specified name on the specified base at the given X and Y position.

Result : QuestDone (: *ShowMessage*)

When the trigger is activated, this will finish the quest and write the achievement (if any) to the player's Quine5000. Optionally, you can have the game show a message that tells the player that the quest has been completed by adding ShowMessage.

Result : RemoveCargo : *cargo type* : *amount*

When the trigger is activated, this will remove the specified amount of the specified cargo from the player's cargo hold. Valid cargo types are the names of the 40 different commodities traded in the game.

Result : RemoveCredits : *amount*

When the trigger is activated, this will reduce the player's credit balance by the specified amount.

Result : RemoveItem : *item name*

When the trigger is activated, this will remove the specified item from play and free up its item slot.

Result : RemoveMission : *mission name*

When the trigger is activated, this will remove the specified mission from the player's Quine and nav computer and free up its mission slot. Remember that you cannot have more than 5 missions in use at the same time – that's why it's important to remove them when they're no longer needed in the quest (also, they clutter the player's list of missions, since the player cannot remove quest missions from the Quine himself).

Result : RemoveNavPointMark : *system name or All* : *nav point number*

When the trigger is activated, this will cause the specified nav point in the specified system to no longer show up as red on the nav computer. If you want to remove the markings on all nav points, just specify All instead of a system name and nav point number.

Result : RenameItem : *item name* : *new item name*

When the trigger is activated, this will immediately rename the item with the specified name to the specified new item name.

Result : Timer : *timer number* : *Start, Stop or Set* : *value*

When the trigger is activated, this will Start, Stop or Set the timer with the specified number (1 - 5) to the specified value. When a timer has been started, it will be increased with 1 each game second. Stopping a timer will not reset it, so starting it again will make it continue from the value it had when it was stopped.

When setting the timer value, you can specify the timer value as X, -X or +X. Timers are unaffected by scene changes.

Result : Variable : *variable name* : *value*

When the trigger is activated, this will set the variable with the specified name to the specified value. The value can be specified as X, -X or +X. You can only use the name of a variable you declared in the beginning of the quest script.

Repeats : *number of repeats*

A trigger can only be activated one time per default, but if you want it to active more than once, you can use this to specify the number of times the activation can be repeated. Specifying the number of repeats as -1 will cause the trigger to be able to activate an infinite number of times.