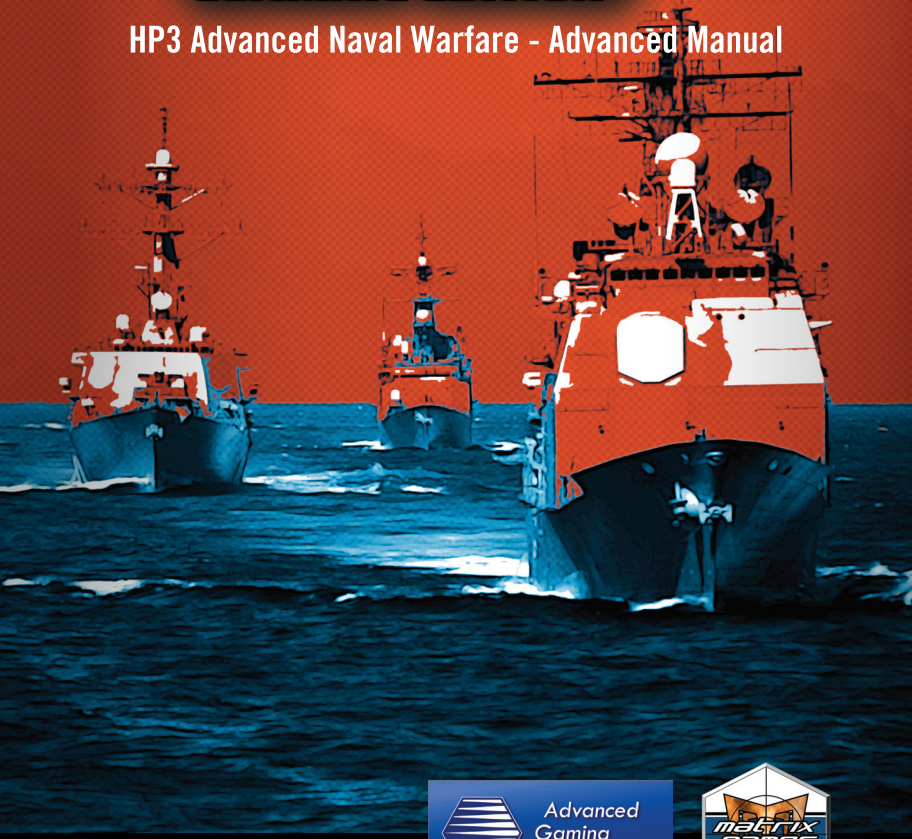
GAME MANUAL

Larry Bond's HARPOON ULTIMATE EDITION™

HP3 Advanced Naval Warfare - Advanced Manual



Advanced Gaming Systems

matrix games

EPILEPSY WARNING

PLEASE READ THIS NOTICE BEFORE PLAYING THIS GAME OR BEFORE ALLOWING YOUR CHILDREN TO PLAY.

Certain individuals may experience epileptic seizures or loss of consciousness when subjected to strong, flashing lights for long periods of time. Such individuals may therefore experience a seizure while operating computer or video games. This can also affect individuals who have no prior medical record of epilepsy or have never previously experienced a seizure.

If you or any family member has ever experienced epilepsy symptoms (seizures or loss of consciousness) after exposure to flashing lights, please consult your doctor before playing this game.

Parental guidance is always suggested when children are using a computer and video games. Should you or your child experience dizziness, poor eyesight, eye or muscle twitching, loss of consciousness, feelings of disorientation or any type of involuntary movements or cramps while playing this game, turn it off immediately and consult your doctor before playing again.

PRECAUTIONS DURING USE:

- Do not sit too close to the monitor.
  Sit as far as comfortably possible.
- Use as small a monitor as possible.
- Do not play when tired or short on sleep.
- Take care that there is sufficient lighting in the room.
- Be sure to take a break of 10-15 minutes every hour.

CONTENTS

17. ADVANCED MANUAL INTRODUCTION

Now that you have mastered playing Harpoon 3: ANW (H3ANW), there's plenty more to explore. The aim of the Advanced Manual is to acquaint the player with the vast creation, customization, and editing features available. In addition, there is some advanced information on naval warfare as it pertains to the H3ANW system for players looking for in depth knowledge on this type of war and the technologies therein.

This manual was originally developed on the AGSI Wiki website by the people listed in the credits. It was then extensively edited by AGSI and Matrix Games.

Current information on the game, errata, hints, etc are found on the AGSI Wiki referenced from http://www.computerharpoon.com

Finally, it is critically important to know that with each release of Harpoon 3, the computer software moves closer to the Harpoon System Models. These Models are created by Larry Bond and Capt Chris Carlson - USNR (Ret). The Models, mostly created in MS Excel, are then implemented in two different mediums. One on paper, that being the "Trilogy Miniatures" rules that include *Harpoon* (modern naval warfare), *Command at Sea* (WWII), and *Fear God and Dreadnought* (WW1). The other being *Harpoon 3 Advanced Naval Warfare*. It helps to understand the model if you can see it implemented in both mediums. As such, this Advanced Manual includes references to the miniatures. If you wish to purchase a copy of the miniatures, you should visit the publisher at Clash of Arms http://www.clashofarms.com/.

The miniatures are referred to by their Editions:

Harpoon 3rd Edition

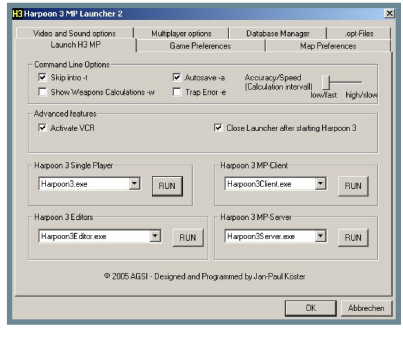
Harpoon 4.1 (4th Edition with revisions)

Harpoon 5 (part of the integrated WW1-Modern *Admiralty Trilogy*)

18. H3ANW LAUNCHER CONFIGURATION

18.1. LAUNCH H3ANW WINDOW

This is the first window you will see when you run the H3ANW Launcher for the first time. It's pretty self explanatory. Most importantly you need to click on the respective "RUN" button to launch that the desired particular H3ANW product (i.e., Game Engines, Server, or Editors).



18.2. COMMAND LINE STARTUP OPTIONS

Skip Intro: If this checkbox is ticked the Intro video will not be shown when H3ANW is started

Autosave: If this checkbox is ticked the game will regularly create a save game as a backup.

Show Weapons Calculations: If this checkbox is ticked the calculations that decide whether a weapon hits or misses will be shown in the In Game Message box

Trap Error: If this checkbox is ticked the game will log additional data used for debugging the application. This should not be needed in regular gameplay.

Accuracy/Speed setting: The accuracy/speed setting sets the maximum amount of time between calculations in the game window. The default for this is once every 15 seconds, so if you run at higher time compression the game engine will only calculate each 15 second step (which is the fastest setting). You can reduce this to 1 second which means that even at maximum time compression each second is calculated. This is much more accurate, but also requires a much better hardware to run at the same time compression.

18.3. ADVANCED FEATURES

Activate VCR: Since H3ANW the game includes a VCR feature that will allow the players to review their scenario after they are done with it. For detailed instructions on the VCR feature please take a look at Section 19.0 H3ANW VCR.

Close Launcher after starting H3ANW: If this box is unchecked, the Launcher application will remain open even after an instance of H3ANW is launched. This is useful if a player wants to run the server and client on the same computer.

18.4. RUN BUTTONS

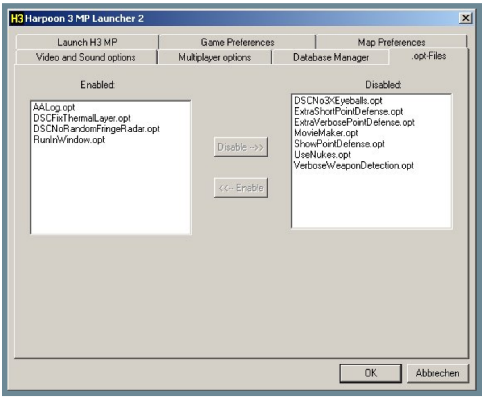
These buttons are used to run the different H3ANW applications. Simply select the version product you want to use from the pull down menu and hit the **rRun** button. H3ANW should then start.

18.5. MAP PREFERENCES

Map Preferences, again reasonably self explanatory, this shows the default selections which can be changed for each map in the game.

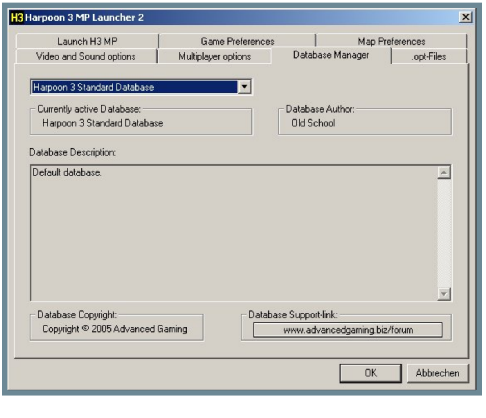
18.6. OPT FILES

Information on the .OPT files and what options they enable/disable can be found in 28.2. .



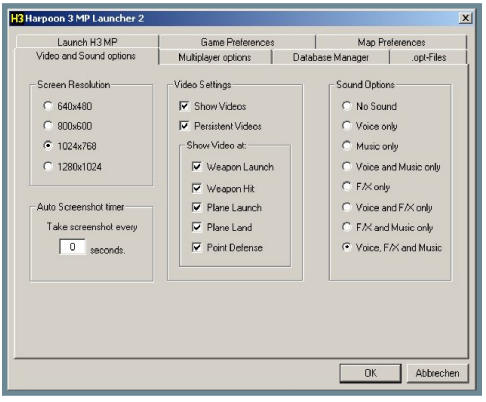
18.7. DATABASE MANAGER

The DB-Manager screen allows you to easily switch between databases. Place each user database inside the database directory and the launcher will automatically display them in this window. The DB-Manager will also adjust the scenario folder to point at the “Scenarios” folder inside the current DB’s folder, so if you want to download additional scenarios place them here.. Additionally, the DB manager will automatically create a battle set file for all standalone scenarios it finds in the scenario folder.



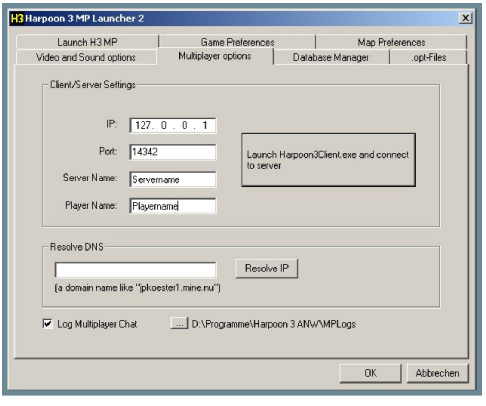
18.8. VIDEO AND SOUND OPTIONS

The Video and Sound options, again, self explanatory.



18.9. MULTIPLAYER OPTIONS

The Multiplayer options screen will allow you to configure all settings needed to run or connect to a Multiplayer server.



IP: In this field please Eenter the IP of the server you want to connect to (you will need to ask the person running it). If you want to run the server yourself you can enter 127.0.0.1 which is the internal IP of your own computer.

Port: This is the port you want to connect to or run the server on. As a default this is 14342 and it usually should not be necessary to change it, unless there are issues with a firewall setting.

> **NOTE: Matrix Games nor AGSI can provide support, or be liable for, settings made to your networking hardware or software.**

Server Name: If you want to run a server, you need to give it a name.

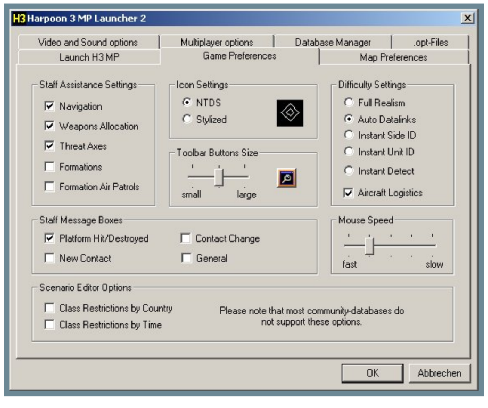
Player Name: Enter a nickname here which people will see when you log onto a server.

Resolve DNS: This is an advanced feature that will allow people to set up static server addresses which the Launcher will then resolve into IPs.

Log Multiplayer chat: The name says it all. Choose the location for the chat logs with the button to the right.

18.10. GAME PREFERENCES

Here you can set the **Game Preferences**. Please refer to the relevant article to find out what the specific settings do.

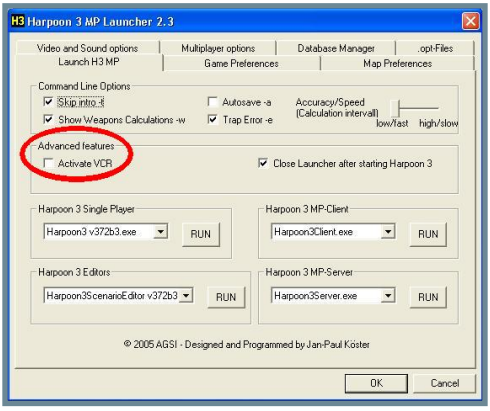


19. VCR

The basic ability to record and play back a game is available in all versions of the system. H3 MilSim offers more advanced capabilities suited for training. A point to note about VCR files is that they are only compatible with the same version of the database in use when they were recorded. For example, If you record a file using XYZ Database version 250, the VCR will only playback with the version 250 of that DB.

19.1. ACTIVATION

VCR can be enabled using the VCR option in the H3ANW Launcher as shown here:



Keep in mind that on slower machines, there will be a *noticeable* performance hit with this feature on and with all machines, it will QUICKLY consume hard disk space.

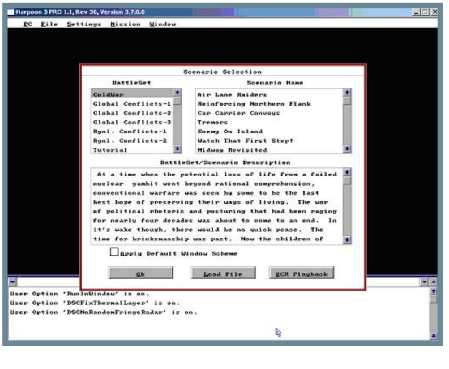
19.2. RECORDING

To record a session, you simply play and complete a game in a normal fashion. Each game session creates a single VCR file as described below. The VCR files are written to your \Scenario directory. The format of the file names is session.001.vcr and each scenario you record after the first will be numbered sequentially (session.002.vcr, session.003.vcr...). Moving the files out of the directory or deleting them will start the sequence again. Please be sure to not modify the names or format of these files.

19.3. PLAYBACK

To playback a VCR file, you must have it in the \scenario folder with the Scenario it was created from. Run the Harpoon exe and click the “VCR Playback” button on the scenario selection window when the application opens.

Just as you select a scenario you will select the .vcr file you would like to view and complete the process by clicking OK. Your recorded scenario will load with a *Playback Control*



Menu and *Viewpoint Selection Menu* superimposed over the typical Harpoon display. These menus can be moved to any part of your window with a click and drag to allow you to set up their appearance as you wish.

The "Playback Control Menu" should look very similar to the interface of your home VCR or DVD player and acts in the same manner with a few unique features.

19.4. BUTTONS AND FUNCTIONS OF THE PLAYBACK CONTROL MENU

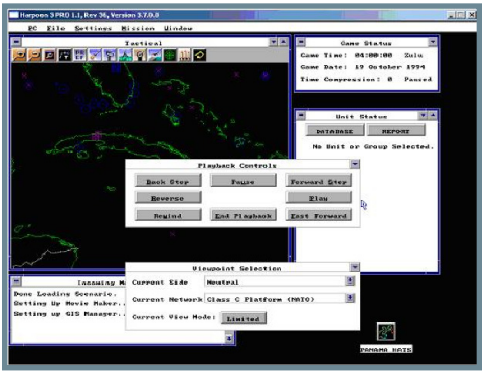
Play - This plays the recording in normal game speed.

Pause - This pauses a recording in any playback setting

Fast Forward - This plays the recording at an accelerated rate of speed.

Forward Step - This plays the recording forward by one time increment.

End Playback - This ends the Playback session.



The "**Viewpoint Selection Menu**" relates to the basic VCR functionality. It consists of two drop down menus and a Show All button.

19.5. DROPDOWN MENUS AND BUTTON OF THE VIEWPOINT CONTROL MENU

Current Side Menu - This drop down menu allows you to view from the side of your choice by click and dragging to the side you choose.

Current Network Menu - This drop down menu allows you to view the recording from the perspective of a unit within your current side of choice by clicking and dragging to the specific unit

Show All Button - Clicking this button allows you to view all units from all sides of the game.

20. H3ANW MULTI-PLAYER

Multiplayer (MP) requires a server; we recommend a 1.5 GHz system with 512mb of RAM or better. Then you'll need one or more clients. The game's AI is offline in the MP products. It is possible for one machine to be both Server and a client, but performance suffers accordingly unless you have a fast dual core system with 1Gb of RAM We also recommend that you have some sort of Instant Messenger or Voice over IP connection going the first time you setup an MP session with a new player. Using a free web conferencing tools is even better as the more experienced player can help configure settings on the other players' systems.

Note it is a courtesy to stipulate the database version and scenarios being offered to another player. The Server can only run one version of the database and should only run scenarios created for that exact version of the database. The Server will push down the database and scenarios to the Clients so they will be consistent.

The illustrations are H3MilSim illustrations; simply ignore the "Umpire" button.

20.1. FIREWALLS AND ROUTERS

Firewalls usually block all traffic from the outside and need to be specifically told that connections coming in on port 14342 should be allowed to pass through to the H3ANW server. How this is accomplished varies greatly from one firewall software to the other and users should consult the various manuals on how to allow this.

In addition to the hardware firewall often integrated in routers (which principally pose the same problem software routers do) routers face an additional problem. When seen from the internet the router has only one IP address although there might be more than one computer connected to it. Because of this the router needs to know which of the connected computers is the target of an incoming connection attempts on port 14342. Some router manufacturers call this setting Virtual Server or Port forwarding and users should refer to the different user manuals of their routers on how this can be accomplished.

In general, to allow your router to operate H3ANW you must do the following:

- Ensure that your software firewall allows H3ANW to establish outgoing connections to port 14342 (client)
- Ensure that your router allows H3ANW to establish outgoing connections to port 14342 (client)
- Ensure that your software firewall allows incoming connections to contact the H3ANW software on port 14342 (server)
- Ensure that your router allows incoming connections to port 14342 (server)
- Ensure that your router knows which computer on the local network should receive incoming connection attempts on port 14342 and forwards all packets to it (server)

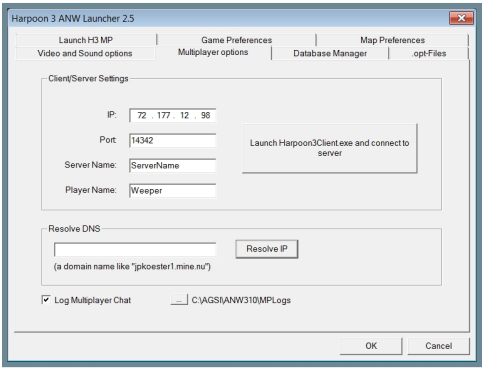
At the command prompt, do an **ipconfig** command and copy the IP address for your active network adapter. That is what the client configurations will need in addition to the server name above. If you choose to run a Client on the Server computer, the loopback address 127.0.0.1 may be used.

If you are behind a router on your LAN you should use your internal IP address for the server. People outside the router (e.g. on the internet) should use your external IP.

NOTE: AGSI can not provide you support, nor be responsible for, any changes you make to your firewall

20.2. SERVER SETUP

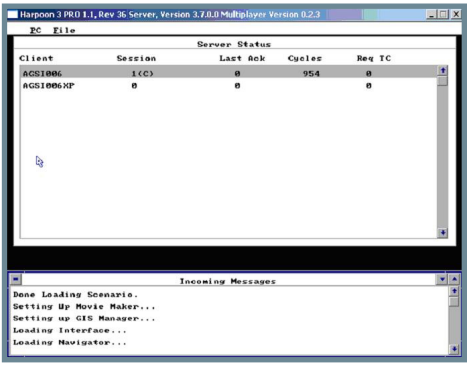
The server is configured via the H3 Launcher under the Multiplayer options.



Setting your IP requires knowing your IP if on the Internet (see firewall section below). Your Server Name is arbitrary, and the Port is as shown (normally the default is used).

20.3. CLIENT

The players will configure their clients to map to the given IP and Server name. The Server must be running before the Clients can connect. There may be a brief delay as the host database is uploaded to each client. Keep in mind that the host may also run a client from the server, allowing him/her to be a player as well. This will cause a performance hit unless the host is running a recent dual core system with a fast hard disk.



20.4. LOBBY

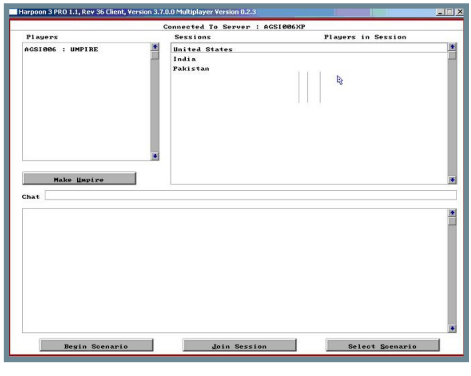
The Clients will now be logged into the main lobby. The lobby allows you to chat and set up games as well as creating a session (i.e. a multiplayer game). You create a session by having one player click the Create Session button, which then prompts the player to name the session. Everyone can then see the created session. They then join via the Join Session button.

20.5. BATTLE SET/SCENARIO SELECTION

The controlling client that created the session will then be able to choose the Battle Set. These are intuitive and similar to the normal H3ANW scenario selection.

20.6. SIDE/ROLE SELECTION

From there the controlling client then assigns each player to a side. Players can be assigned to any side within a scenario and can also be assigned to a side called Spectator. This side allows players to view the game from the sidelines, offering a number of viewpoints to watch from. However, a spectator may not issue orders to any units. This also precludes the spectator viewing dialogs from which such orders may be given, such as the Launch/Ready Aircraft Dialog. Once side selection is completed the controlling client initiates the game and the game begins.



20.7. BASIC GAMEPLAY

Game play is very close to H3ANW Single Player. All games will be played to their duration unless terminated early by clients or server. The victory screen will appear as normal, displaying game results. You then have the option of exiting the game or returning to the lobby for another session.

20.8. CHANGES IN PLAY

Time Compression is handled differently in the MP products than in Single Player H3ANW . Basically it operates on a veto basis. The lowest setting of any human player (see AI players below) is the value that is used. This does introduce the artificial aspect of someone decreasing the Time Compression in order to plan and launch an attack or just to unnerve an opponent. The Game Status window in MP has an additional Time Compression display showing your

requested time compression, below the current time compression. The procedure for creating a mission in the *Mission Editor* has changed slightly. You will notice that you do not have "Edit Now" option available in your Mission Editor Menu. The reason why is that the game will need to send an update prior to you adding the unit to properly implement the mission. As such after you create the mission and exit out of the *Mission Editor* you will have then "Edit" that mission to assign platforms to the mission. Missions function as they always have so there is no change in behavior just procedure.

20.9. ADDITIONS TO PLAY

MP offers a built-in Chat feature. This is accessed by pressing control plus the C key (Ctrl + C). A small chat console will then appear allowing you to type a short message and send it by pressing okay or enter. This message will appear for all clients and in the Server's message window. At the bottom of the chat window is a small checkbox that will allow you to either speak to all players or just your own network. In this case your network is any side that is allied to you. The box is not checked by default and you will want to make sure the box is checked or un-checked depending on what information you want to pass on.

The Spectator Feature provides a new dialog box that allows the Client to select which Side they will see – or to choose ALL sides.

20.10. OTHER CHANGES FROM SINGLE PLAYER HARPOON

The nature of MP play means that there is going to be a higher load on the program engine to accomplish the same things. This means that single player games will run slightly slower in MP than they would in single player ANW. To this end AGSI recommends that dedicated MP scenarios be no larger than 500kb in size.

21. TECHNICAL NOTES

You speak of aircraft carriers and of the construction of new types of ships...while at the same time completely ignoring the economic situation of our country and the corresponding conditions of our technical means... -V.I. Zov, Soviet Naval Commissar, addressing the Soviet Naval Academy

21.1. PROPULSION

While some of the weapons and platforms in H3ANW share common propulsion systems, most do not. To simulate the vastly different operating parameters and performance of all these different engines, boosters, batteries, rockets, etc. an extensive database on all the propulsion systems is used by H3ANW. The name and type, as well as valid fuels and consumption rates are tracked for each system. Each propulsion system may perform differently at different operating altitudes, each offering different speeds and consumption coefficients. Every mobile platform in the simulation contains a propulsion system (or perhaps more than one) and fuel type. These propulsion systems know what type of fuel they can utilize and how fast they will consume it. These layered models allowed us to simulate UNREP (underway replenishment) and air-to-air refueling.

In one test run, we found a ship with a voracious gas turbine engine, which, when all of its fuel was gone, started in on the fuel stores for its embarked helicopters. The gas turbines used by this ship really were capable of efficiently burning almost anything, and thought the helo fuel would make a lovely dessert.

21.2. WEAPONS

One of the trickiest pieces of early modeling in H3ANW was the employment of semi-active radar homing missiles (SARH) which, due to engagement geometry, were forced to share illuminators. It was a simple enough problem when confined to illuminate-all-the-way missiles, but several real missile systems had inertial/command guided fly out and only required illumination in the terminal phase of their flight. Thus, the simulation keeps track of when each director would be busy (in the future) and prevent a missile from firing if no directors would be available at the intercept time. So each director/illuminator got a rolling bit field representing a 15 second engagement slice of time and we wrote a simple timesharing system to keep the launchers honest.

Considering that each mount has a rate-of-fire (ROF) cycle and finite reload time, the resulting complexity has grown beyond the user's ability to effectively micro-manage. In real life, a trained Air Warfare Officer and his crew will have a very hard time keeping up with the multi-dimensional geometry of even the simplest engagements. Since the average H3ANW player would probably have even less training, the computer simply had to handle most of the functionality. The design paradigm of H3ANW states clearly that "Once the missiles start to fly, the user is mostly out of the loop". While this may be less fun than shooting each missile yourself (a la most flight simulators), the emphasis is really on learning about and employing modern naval tactics, and not how quick your mouse hand reflexes are.

In real life, a semi-active radar homing (SARH) missile tracks reflected microwave energy that has bounced off its intended target. This energy usually originates from an 'illuminator'. An illuminator is a generally a radar emitter that produces a tight microwave beam used to guide a SARH missile. A typical missile engagement using SARH missiles would probably go like this: search radar will detect a target and tell the illuminator where to point. The illuminator then emits a microwave beam which reaches out to the target. Some of the energy bounces off the target and is visible to the radar seeker head in the missile (which is still on the rails). The missile signals to the fire control computers that it can see the reflected energy. The missile is fired towards the target and will steer itself towards the reflected energy, eventually intercepting the target and exploding. If the illuminator is turned off for some reason, the missile will no longer see any reflected energy and will miss the target. Some missiles will self-destruct when they lose their illumination. There are some problems with this system. First, the illuminator must be constantly pointed at the target during the entire flight of the outbound missile. This limits the quantity of targets to the number of directors available.

Some missiles and torpedoes employ search patterns in the terminal phase of delivery. This means if they do not see an appropriate target when they have reached their activation point, they will alter their course to search for one. Missiles and sub-launched torpedoes will initiate an expanding 'snake' pattern, which makes the weapon simply zigzag back and forth every few minutes by about 45 degrees left and right of its base course. In contrast, aerial delivered torpedoes launched at uncertain submarine contacts will almost always begin an expanding circle pattern (a spiral), occasionally changing depths in its search for a submarine. Care must be taken not to launch at an uncertain target in an area where friendly or neutral platforms might be detected by the weapon's seeker head. The weapon cannot tell friend from foe or combatant from civilian and will engage any platform matching its target parameters.

Air-delivered gravity bombs may be tossed ballistically. This means that the higher and faster an aircraft is flying, the greater the distance from the target the bomb can be released. This allows aircraft to avoid heavy concentrations of point defense weapons (like handheld SAM launchers and small arms fire).

21.3. MAPS

21.3.1. MAPS 101

Latitude is North and South starting at the equator and ending at the poles. Longitude is East and West starting at the Greenwich Meridian (London) and ends at the International Date Line.

Along any great circle (any line of latitude, the equator itself, or any other plane which includes the earth's center) one Degree of Arc is 60nm. Thus one degree of arc along the equator going East or West is 60 Nautical Miles, and one minute of arc is one Nautical Mile. One degree of arc in the true North or true South direction is also 60 Nautical miles, and one minute of arc is one Nautical mile. As one moves away from the Equator and closer to the poles one minute of arc in the East or West directions becomes less than one Nautical Mile and approaches a zero value at the poles.

As already mentioned one degree is broken down into 60 minutes. In addition each minute is broken down into 60 seconds. Therefore one minute equals one nautical mile (along any great circle) and one second equals about 30 meters.

Having this understanding in mind, along with the different map projections, will enhance the player's simulation experience as a player or scenario designer.

21.3.2. MAPS IN H3 ANW

The maps in H3 ANW are authentic in their visual display. They are based on how a variety of actual tactical systems display topographic and bathymetric data. Additionally, vector maps offered free scaling and sizing without any data loss, and provided support for map projections. This approached addressed two of the major map limitations in the original Harpoon.

The maps are drawn from the following sources (circa 1995); The Defense Mapping Agency, The Central Intelligence Agency, The National Oceanic and Atmospheric Administration, The National Geophysical Data Center, The Governments of Great Britain, Canada, and Australia. The polygon data represented coastal points on a 1000 yard scale for the entire planet and topographic and hydro/bathymetric data points every 30 miles (at the equator).

21.4. DECOYS

"All warfare is based on deception. Hence, when able to attack, we must seem unable; when using our forces, we must seem inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near. Hold out baits to entice the enemy. Feign disorder, and crush him." - Sun Tzu, The Art of War

21.4.1. PLATFORM DECOYS

A decoy is any apparatus (generally a mortar type weapon) that is used to attempt to trick the seeker-head of a missile into thinking its target is elsewhere. They are considered part of Electronic Warfare.

Decoy deployment is generally an operation that happens throughout the duration of an engagement. A decoy launcher can have several functions. Firing at long range (either while enemy aircraft are approaching launch range or while the missiles are inbound) can provide the targeting radar/missile seeker with multiple targets. If all targets are engaged then the attack is diluted, reducing the number of weapons that are targeted on actual platforms, which in turn reduces the load on the ships defensive systems (i.e., SAMs, Guns, CIWS, etc.).

In reality the usual counter-measure is called **'Moving Target Indication'** (MTI). Radar distinguishes a real target from the decoy (in this case chaff) because the real target is moving so much faster. However, such a distinction is harder to make when the target involves a ship due to the smaller relative speeds.

The second function of the decoy launch is called **seduction**. This is accomplished by firing a large 'cloud' of decoys away from the ship to present a new and larger target for the seeker to attack. Generally an intermediate range function, it's most useful just before the seeker head of the missile turns on.

Note that *range gate* is related to tracking. A sensor in track mode looks within a series of gates (range, speed, and bearing) for a known track. If the contact is NOT detected within the gates, a wider area must be searched to re-establish the track. Breaking track and forcing a re-acquire re-track makes all sensor targets (including decoys) within the larger gate a potentially valid target. Range gates do NOT alleviate the effects of chaff.

Finally, there is a last-ditch tactic, **centroid-seduction**. This missile's seeker will generally home in on the centroid of the target's radar return, seeing the ships target as a complex array of point and corner reflectors. If a large decoy cloud is added around or near the ship, then it alters the shape of the radar return in such a way that the missile will adjust its course to pass through the new centroid and (hopefully) fly overhead of pass to the side of the targeted vessel. Success in this mode depends on the reflective cross-section of the decoy cloud compared with that of the ship proper. If the cloud is designed to move the centroid up then that may be countered by having a radio altimeter that keeps the missile at low altitude.

Chaff (and IR Decoys and Flares) may be launched by rocket or mortar. Rockets provide longer range (for the dilution role) and do not impose any load on the surrounding deck. However, they are relatively large, and the naval architect must protect against their back-blast. Mortars, and their ammunition, are more compact. Both types are commonly used.

21.4.2. MODELING NOTE: DECOYS IN HARPOON 4TH EDITION MINIATURES

Decoys in the *Harpoon 4 miniatures* function as a reduction in the Probability of Kill (PK) verses the attacking missile. They operate along a generational line similar to self protection ECM (i.e. ALQ-131).

21.4.3. DECOYS IN H3 ANW

Using the same generational modeling system as implemented in the Harpoon miniatures, decoys are considered a point defense weapon. They are (and can be) launched and expended like any other weapon but only factor into the weapon resolution calculations at the moment

of impact. Decoys can be launched at any time but they will have no effect until (and if) the weapon reaches its assigned target. They function in the same manner as in the miniatures rules in that, they provide a reduction to the PK of the attacking weapon. This reduction or penalty (and any others) is then applied to the final PK of the weapon during the random number generation at the point of impact. (i.e. the roll of the dice). Decoy's are not visually represented, but are counted as stores and can be depleted like any other ordinance.

21.4.4. DECOYS IN THE HARPOON 3 ANW DATABASES

Decoy performance is based on the values assigned to them in the ATA and Surface PK fields in the database.

Acoustic (torpedo) decoys use the Near Surface Atk PK value.

Radar decoys (chaff) use the Near ATA Atk PK value.

Infra-Red decoys (Flares) use the Far ATA Atk PK Value.

The remaining Far Surface Atk Pk database value is not utilized..

Values in these fields must be between 0 and 99 and are a function of a percentage reduction of PK. All fields must be filled with a 0 representing no affect against that type of weapon while a value of (for example) 15 will represent a 15% reduction in the PK of the approaching weapon.

The value must never be negative as this will result in an abnormally high reduction in the PK of the weapon (100%, making the incoming weapon effectively useless and tilting the balance of the game). This applies to all EW sensors and decoys in game. This is the responsibility of the database editor.

21.5. ELECTRONIC COUNTER-MEASURES (ECM)

"A ship in port is safe, but that's not what ships are built for."
- Grace Murray Hopper

If a ship or aircraft is equipped with ECM equipment, it can be used to degrade enemy sensors. Select Active from the Sensor dialog box to activate the ECM. Remember that in many cases, using ECM will degrade your side's sensors as well. When a unit or group is being jammed, a small "X" will appear in the upper-right side of the unit or group symbol. If a unit or group on another side is using jammers, a small "Q" will appear in the upper-right side of the symbol for the unit or group using the jammer.

ECM (now called Electronic Warfare in the US) falls into two broad categories: **noise jamming** and **deception jamming**. Noise jamming seeks to blot out the signal in a manner similar to shining a flashlight in your eyes. It can cover a broad spectrum (called barrage jamming) or specific frequencies (called spot jamming). In real life, the best counter to spot jamming is frequency agility which gives the jammed sensor the ability to operate outside the frequencies that are being jammed. This is made harder at higher frequencies (such has the NATO K-Band [the US Ka and Ku Bands]) because the higher the frequency, the broader the range of frequencies that the jammer must cover.

Noise jamming is generally used to screen formations of aircraft and ships in which a jammer will generate enough power to make it impossible for the other side to distinguish a nearby High Value Unit (HVU).

Deception jamming generally depends on a more involved analysis of the radar to be jammed. Because the jammer must send back a false echo, then systems on board the jammer must be able to monitor the Pulse Repetition Frequency (PRF), scan rate, and pulse width. The idea being to send back signals that will fool the radar into creating a false contact either by altering the range (range-gate pull off), speed (velocity-gate pull off), or bearing (by sending a signal down the radars side lobes). This is mainly a battle between hardware sophistication and its associated software.

21.5.1. MODELING NOTE: ECM IN HARPOON 4TH EDITION MINIATURES

The following notes are specific to the 4th and 5th Editions of the Harpoon models.

Because of the complex and highly classified nature of ECM, *Harpoon 4th Edition* uses a simplified approach, building in the effects of Radar Counter Measures into weapon PK values and radar detection capabilities. Although some information is available for many of the systems in use, knowledge of their combined use is limited, and the effects of both sides using chaff, jammers, deception repeaters, and flares all at the same time can only be guessed at. The emphasis in *Harpoon 4th Edition* is simplicity, and given the accuracy of unclassified information available, this approach gives a reasonably accurate approximation of the effectiveness of electronic or acoustic countermeasures.

ECM actions (turning on jammers, deploying chaff) must be ordered in the plotting phase of a turn. Exact employment tactics are not necessary as it's assumed the countermeasures are used correctly.

In H4.x Master Rules play, standoff jammers like ALQ-99 will automatically degrade all radars by a certain percentage of their range (typically 50 to 75% depending on the generation of the system and how many pods [jammers] there are).

Self-protection jammers will apply a PK reduction similar that used by chaff or flares. Users are advised that to consult the *Harpoon 4th Edition* Master Rules (specifically pages 6-31 and 6-32) for further information on Electronic Warfare.

21.5.2. ECM IN HARPOON

Harpoon 5 paper rules are utilized for ECM in Harpoon 3 ANW. A non-cooperative jammer within the radar horizon (defined as not being on the same network as the detector) reduces detection range by 75% (25% for radars with the Phased-Array flag set). Passive Radar Detection (RD) on radars is applied as a reduction of the range effect; in percent (i.e. Passive RD of 10 reduces ECM effect to 65% / 15% range reduction).

A 'non-cooperative jammer' is defined as any jammer on a side that is either neutral or hostile to the side that the user is currently playing. H5 rules currently abstract ECM; their reasoning is explained in section 6.7 of the H4 rules. (See the first paragraph in the above section, 29.1 Modeling Note: ECM in *Harpoon 4th Edition* Miniatures

Therefore, all radars being jammed will have detection ranges reduced by 75% of their range. Radars with the phased array flag activated will have this reduced to 25%. Furthermore, the DB Designer will have the option of fine tuning that by using the Passive RD field. The number specified in the field will be a percentage increase in detection range.

Phased array radars should have the 'Phased Array' flag set in the database.

Therefore:

SPS-49(v)5 2D Air Search Radar Peak Power: 360kW giving an Output of 552 Detection ranges of: 338/213/159/68/20nm for an average input value of -2136

If Passive RD is 0 when the radar is jammed then the detection ranged will be reduced to 25% of those numbers above. Any value above 0 in the Passive RD will increase the detection range by that value.

Therefore:...

If Passive RD = 10 then detection range is reduced by 65%, 20 results in a detection reduction of 55% and so on.

Also keep in mind the effect of the radar horizon on jammers. This will significantly reduce the capability of shipboard jammers against low flying aircraft and missiles because these platforms will be flying below the jamming beam. On the other side of the coin airborne jammers like the EA-18 Growler will be able to control the radius of their jamming by simply changing altitude. Please check the H3 ANW Platform Workbook for details on the radar horizon. Please see the AGSI Wiki for the Platform Workbook.

It's important to note that ECM affects the detection range only and not to the cross section value of the target being searched for. So if a radar would normally pick up a target at 100nm then under jammed conditions it would pick it up at 25nm. Phased arrays would pick up the target at 75nm and a radar with an ECCM rating of 25% would pick up the target at 50nm (75% - 25% = 50% range reduction). Either way, maximum jamming range is set by the radar horizon of the jamming platform.

21.5.3. ECM AND DATABASE EDITING

In game ECM has no range other than the radar horizon, and the game engine will not observe the maximum range field of the sensor annex. At this time the user will have to calculate manually, based on altitude, the maximum effective range of the jammer. We can only guarantee that platforms OUTSIDE the range ring will not be jammed.

Database authors are also reminded that all values in the output values for the jammer (and radars too) must be positive. There will be strange weapons effects (100% effectiveness of countermeasures) should the value be negative. This also applies to the ATA and PK values of decoys.

21.6. ELECTRONIC SUPPORT MEASURES (ESM)

"Men of sense often learn from their enemies. It is from their foes, not their friends, that cities learn the lesson of building high walls and ships of war." - Aristophanes

ESM involves the passive reception of enemy electromagnetic emissions. By processing these emissions against a database of known emissions, and comparing the frequency, pulse repetition rate and other details too extensive to cover here, modern ESM suites like the SLQ-32 are capable of identifying the class of emitter which produced the transmissions.

A unit which radiates, and is subsequently classified by the enemy, is not necessarily targeted or even located with any degree of certainty. If a single unit in the force receives the transmission, a line of bearing to the source is generated. If multiple units in the force receive the intercept,

they may correlate their bearings to define an ellipse known as the area of probability, within which the radiating unit must be. This technique is known as passive cross-fixing. The greater the physical separation between the units coordinating for the passive cross fix, the greater their bearing separation will be and the more accurate and narrow the area of probability.

Electronic Support Measures (ESM) are means of locating and identifying an emitter passively, e.g., to target a weapon (in theory the target identifies itself). For example, an anti-radar missile (ARM) uses ESM guidance. ESM systems concentrate on identifying a particular emitter. Their relatives, radar warning receivers (RWR), concentrate instead of the quick warning of likely threats. A very precise ESM system designed to measure emitter parameters may be called an Electronic Intelligence (ELINT) system. Current US official parlance has replaced ESM with 'Electronic Support' (ES).

For ESM systems the first great divide was between precision and wide-open types, both in direction and frequency. In both cases, precision equates in higher gain but also to a lower probably of intercept since the narrow frequency/direction window may not be pointed in the direction of the threat.

ESM receivers fall into two categories: Superheterodynes (SHRs) and Crystal Video Receivers (CVRs). SHRs scans a narrow tuning window (and as a result are more sensitive) through the systems frequency spectrum thus both detecting incoming signals and their frequency fairly quickly. The trade off is that it can't scan quickly enough to pick up a short signal such as a radar pulse. As such they are used usually in the Communications Intelligence (COMINT) role.

CVRs detect all signals over it's bandwidth but is relatively insensitive. In the early 60's aircraft RWRs used CVRs because they were sure to detect strong signals of any frequency (within the coverage area) pointed continuously at the detector, such as those from Fire Control Radars (FCRs). CVRs would be far less likely to detect search radars but they were much less important. Later, it became possible to make CVRs sensitive enough by adding front end amplifiers. Because it is a broadband receiver, a CVR can identify a signal by frequency band, but it cannot tell where in the band that signal is located.

If signal characteristics are broadly known, a system can trade off size (for instant directivity) against time (to achieve directivity using a small antenna), using a technique called super-resolution.

Some emitters are unique to a single class of ship, aircraft, or submarine; as soon as that emitter is detected, the commander will be able to classify the threat exactly. Most emitters, however, are carried on numerous platforms. As such, a single emission may only produce a list of possibilities as to the originating platform. When this occurs, prudent commanders must assume the worst possible case of all the potential threats. Subsequent emissions of other radar types from the same bearing may allow one to narrow the list of possible threats, by excluding those platforms which do not have both emitter types, but formations of many vessels may still make this evaluation unreliable. Indeed, it is possible for far-sighted commanders to radiate emitters on two separate ships in proximity, possibly leading the enemy to believe a third, stronger vessel is present.

One thing should be readily apparent: For ESM efforts to be effective, the enemy must cooperate by radiating their emitters. Given the potentially deadly effect of a passive, over-the-horizon missile attack, in which the first warning is illumination by missile seeker heads in their terminal phase of flight, one might reasonably question if it is ever worthwhile to radiate and risk this possibility. The answer is yes, but radiating must be done when it is tactically advisable to do so and avoided when it is not. As the battlefield is a dynamic environment, the situations which dictate changes in emission posture are also fluid.

21.7. RADAR

Radar is an acronym for radio detection and ranging. It is a system used to detect, range (determine the distance of), and map objects such as aircraft and ships. Strong radio waves are transmitted, and a receiver listens for any echoes. By analyzing the reflected signal, the reflector can be located, and sometimes identified. Although the amount of signal returned is tiny, radio signals can easily be detected and amplified.

Radar radio waves can be easily generated at any desired strength, detected at even tiny powers, and then amplified many times. Thus radar is suited to detecting objects at very large ranges where other reflections, like sound or visible light, would be too weak to detect.

21.7.1. BASIC PRINCIPLES

Radar operates in a manner similar to a flashlight. Electromagnetic waves reflect from any large change in the dielectric or diamagnetic constants. This means that a solid object in air or vacuum, or other significant changes in atomic density between object and what's surrounding it, will usually reflect radar waves. The reflection is then 'seen' by the radar antenna and processed to show the picture on the radar scope. This is particularly true of electrically-conductive materials such as metal, making radar particularly well suited to the detection of aircraft and ships.

In effect the flashlight is the transmitter and the light that you see on the wall (its reflection) is what you see with your eyes. Radar antennas accomplish both operations instantly.

Radar waves reflect in a variety of ways depending on the size of the radio wave and the shape of the target. If the radio wave is much shorter than the reflector's size, the wave will bounce off in a way similar to the way light bounces from a mirror. Early radars used very long wavelengths that were larger than the targets and received a vague signal, whereas modern systems use shorter wavelengths (a few centimeters) that can image objects as small as a loaf of bread.

Radio waves always reflect from curves and angles, in a way similar to glint from a rounded piece of glass. The most reflective targets have 90° angles between the reflective surfaces. A surface consisting of three flat surfaces meeting at a single corner, like the corner on a block, will always reflect directly back at the source. These so-called corner cubes are commonly used as radar reflectors to make otherwise difficult-to-detect objects easier to detect, and are often found on boats in order to improve their detection in a rescue situation. For generally the same reasons objects attempting to avoid detection will angle their surfaces in a way to eliminate corners, which leads to "odd" looking stealth aircraft.

Radars are generally described by the frequency bands, peak power, pulse width, and Pulse Repetition Frequency (PRF). Frequency (which is inversely proportional to wavelength) determines how precise the radar beam can be, since beam width (in radians) is approximately equal to the inverse of the antenna size in wavelengths. The larger the antenna, the narrower the resulting beam produced will be. Moreover, the larger the antenna the higher the gain; the larger the fraction of the returning echo the radar will be capable of detecting. Frequency (or wavelength) also determines the kind of technology that is required to produce a particular radar signal, and the extent to which that signal will be absorbed by the atmosphere due to environmental conditions (e.g. rain).

21.7.2. HOW TO DEFEAT THE AEGIS SYSTEM

In Ancient Greece mythology, 'Aegis' is the name of Zeus' shield. Forged by Hephaestus, it was an invulnerable shield with Medusa's head at its top (or in the middle depending on the version you believe).

In the 1980s, the United States commissioned a new series of ships fitted with a new micrometric-wave radar called SPY-1B. Ticonderoga (CG-47) and Arleigh Burke (DDG-51) class ships are fitted with this system to protect against large-scale missile threats, most notably from (at the time) the Soviet Union and their long-range bombers equipped with the dreaded AS-4 or AS-6 anti-ship missiles.

The Aegis system can be fully automated or partly automated, according to the environment. The SPY-1 radar guides missiles (either the SM-1 or SM-2, or their different variants) towards their targets (missiles and aircraft) with great accuracy. As in Greek mythology, Aegis was conceived as an invulnerable defense system.

It can, however, sometimes be defeated in Harpoon 3. Here we won't speak about an easy solution that would consist in having more attacking missiles than defending AEGIS-guided missiles. There's another solution which demands a little more effort.

Imagine a scenario where China's navy is battling the U.S. Navy. The U.S. Navy has a carrier group and an amphibious squadron heading towards Chinese-occupied islands. China wants to prevent this and mounts an attack against this force. The U.S. has one carrier, three amphibious ships, and an escort of three Ticonderoga class and three Arleigh Burke class ships. These six ships may be seen as an impervious screen, but it is one that can be pierced.

The best solution is to mount a multi-directional attack on the U.S. ships. To this end, the Chinese need to know the distance to the target, the speed of aircraft (sometimes different types with different speeds), and then it is just a matter of doing the math:

The Chinese will attack from five different bases:

- NW: 706 miles (8 Su-30 + 8 H-6 Badger)
- W: 507 miles (8 Su-30 + 8 H-6 Badger)
- Hainan: 646 miles (8 Su-30 + 8 H-6 Badger)
- Spratly North: 175miles (12 JH-7)
- Spratly South: 190 miles (12 JH-7)

Let's look at the cruise speed of our aircraft:

- An Su-30 and a JH-7 both travel at 540mph, that's 9 miles/ minute.
- An H-6 Badger flies at 459mph, that's 7.65 miles/ minute.

This means that if you want half of these planes (due to a formation having a maximum number of aircraft limit of four) striking at the same time (let's say 95 minutes after the scenario begins to allow all aircraft to take off), your timing should be as follows:

- NW: Su-30 at +17 and H-6 at +3
- W: Su-30 at +39 and H-6 at +29
- Hainan: Su-30 at +24 and H-6 at +11

» Spratly North: JH-7 at +76

» Spratly South: JH-7 at +74

In this manner, all of your aircraft should arrive more or less at the same time. Most of the missiles should be downed by Aegis-guided missiles, but some will make way to their intended targets.

21.7.3. SOME HISTORY ON AEGIS

Keep in mind that Eastern Bloc forces had a large superiority over those of the Western Bloc during the 1970s. Few people realize how close it really was back in the early 1980's; there are several books written on East/West military levels during the Cold War that support this. In the early 1980's, before the U.S. Army fielded new systems (M-1, M-2, Patriot, AH-64, UH-60, etc.), it was a given fact that NATO probably would not be able to hold Western Europe through conventional means, and that tactical nuclear weapons would most likely be needed to halt a USSR-led invasion. However, by 1985, the situation had stabilized and began to tilt towards NATO's favor. The year of decision was around 1982-3. It was during those years that, had the Warsaw Pact attacked, the result would have been a toss up. This is best illustrated in General Sir John Hackett's book, *The Third World War: August 1985*, in which NATO forces were hard-pressed to hold out against a Warsaw Pact attack. It was only the good timing of a B-52 strike that stopped Soviet armored spearheads just short of cutting West Germany in half. Fiction I know but written by men who were there.

If anything, the trickle of increased technology is much slower in naval circles than in any other military arm. Warships are a complex system in and of themselves and all too often cannot support a new technology; a new platform to support it usually needs to be developed. However, placing new technology onto new platforms is much more expensive than placing newer technology onto older platforms. This is one of the reasons why DLGN-38 was cancelled and AEGIS sat in someone's backyard for 10 years while the weeds grew and the dust gathered.

In 1985, VLS was just a drawing on a paper napkin in a restaurant. In 1985, the Mk26 fired two missiles every 10 seconds and the missiles themselves required manual preparation The Mk26 Mod 2 (64 missiles of varying types) fired 32 salvos at 10 seconds a salvo. This means that both Mk26s on a Ticonderoga class CG will empty their magazines in a little over five minutes. SS-N-19 Shipwreck missiles have a speed of Mach 2.5 (approx. 1650 kts) and a range of 300 nm. That means, if launched at maximum range, the Shipwrecks will arrive over their target in 10 minutes.

But that's not the really bad news. The standard SM-2 missiles of the day only had a 40 nm range (SM-2 ER Block 3B was developed for VLS). So if you factor in that the computers on the Ticonderoga will adjust the launch time so that the first salvo tries for an intercept at that 40 nm max range that, gives the Ticonderoga only an extra minute plus the time it takes the SS-N-19 to travel the 40 nm to the target. All told, this means that the Ticonderoga has 146 seconds from the time the first salvo of missiles is fired until the Shipwreck is on (or rather IN the target). In that time, the Ticonderoga will have launched 28 missiles (14 salvos) and that's just against one missile.

Mind that the ship's computers will probably attempt to allocate more targets than what we are talking about here. This works both ways, because it allows more missiles to be engaged with the same Pk, the down side being that the system cannot engage very many targets a second or third time. This is where priority comes into effect. The AEGIS computer will attempt to determine which missiles should be engaged and which should not.

22. IN GAME COMBAT

In Harpoon3 ANW, the simulation is defined by several actions, which occur in the following sequence:

- Search
- Detection
- Launch and Attack
- Defense
- Resolution

22.1. SEARCH

Search is the function where the units attempt to locate an objective or target. It is a sensor battle and can include active (radar or active sonar) and/or passive (ESM or passive sonar) sensors. Finding the enemy first means firing first, that is the single most important concept in modern naval warfare.

22.2. DETECTION AND LOCALIZATION

Detection is a game calculation that's based on several values in the database. For radars and sonars, detection depends on the 'Search Input' (and 'Track Input' for radars only) values. The value is inversely exponential in nature, that is the lower the number the greater the ability of the sensor to detect a target. In the literal sense, the input value rates the ability of that sensor to pick a target out of the background clutter.

For passive sonar and ESM, detection is based on the 'Passive Input' value and it is also inversely exponential.

The in game terminology (for the older manuals) calls the Search Input value the 'Active Detect RD' and the Passive Input value the 'Passive Detect RD'.

22.3. LAUNCH AND ATTACK

Launch is the point in the game where the player (or AI) decides to launch weapons. From the game perspective, a weapon is anything detached from a platform for the express purpose of destroying another platform. Firing weapons requires that the launching platform meet the requirements of the launch envelope for the weapon, and that the target is within the movement envelope of the weapon Launch parameters are defined by target range (in nautical miles), launching and/or target altitude (in meters), and launching and/or target speed (in knots). The launching platform and target must meet all parameters before the weapon is launched, if the parameters are not met, then the player will receive a message stating what the parameter that is out of range in the weapon selection attack dialogue.

Weapons fall into two categories for launch, active homing weapons like the AGM-84 Harpoon missile, and passive homing weapons like the AIM-9 Sidewinder. Active seeking weapons have their own signal generating sensors to seek out the target. They can be launched without the assistance of other sensors on the ship and are Bearing Only Launch (BOL) capable. Passive

seeking weapons are different in that their seekers must be able to deted the target's own native electromatic emissions prior to launch. In the case of the Sidewinder missile, the seeker head is looking for engine heat.

DB Authors are to note that the seekers of passive homing weapons (IR and ESM) should have the passive input value of their seekers set to a value that will detect targets at a distance that at least meets the maximum range (Rmax) of the weapon. It's important to note that in many cases the range of the seeker will exceed the range of the missile itself so this really isn't a violation of realism. Indeed during the Gulf War of 1991, A-10 pilots used the seeker of the AGM-65 Maverick (both EO and IR) as a makeshift FLIR in flight. This runs concurrent with their method of operation where the pilot must select the target and have the seeker lock onto it.

The seeker of passive seeking weapons MUST see the target before they will be able to launch against it. After launch, weapons that have datalinks (either a radio or fixed link like a torpedo wire) attached to it can be controlled to a limited extent by the player. We say limited because a weapon that is still in search mode can be controlled by the player. Using the navigation command (F3) the player can guide a weapon towards the likely location of contacts until the weapon locks on. At that point, control of the weapon is lost and you're committed to attacking the selected target.

The weapon will proceed to its target at cruise speed (set in the propulsion annex of the database) oblivious of anything that happens around it. If it is detected in this transit phase, it can be attacked. These engagements will follow the Harpoon model rules that declare attacking missiles have an Air To Air (ATA) rating of 0 (non-evading targets).

The transit phase of active seeking weapons terminates at it's seeker activation point. Called the 'Pre-Planned Activation Point' (PAP), this is the location where the missile turns on it's active seeker (active radar/sonar). The location of the PAP is automatically set at launch by the game engine based 5 nm from the target or 15 seconds of travel distance, whichever is longer. This means is that a weapon must be traveling at a speed of Mach 1.88 (1205 knots) or greater for it's seeker to activate at a range of greater than 5nm from the target.

Example PAP Ranges

AGM-84 Harpoon 545 knots cruise speed = 2.27nm in 15 seconds. 2.27 < 5 therefore PAP is at 5nm

Kh-22 Burya [AS-4 Kitchen] 1604 knots cruise speed (low end) = 6.68nm in 15 seconds 6.68 > 5 therefore PAP is 6.68nm

The player can alter PAP location of data linked weapons (after launch) by altering the location of the end of the weapons transit time. This is accomplished by using the Navigation (F3) command.

22.4. DEFENSE

Defense is any action taking by a platform being attacked to defend itself. While it usually involves firing weapons, it can also involve certain types of decoys and ECM.

Missile defense consists of a combination of forcing the missile to lose contact with the platform (by using ECM), fooling the weapon into attacking another target (by using decoys), or destroying the weapon (using guns or missiles).

Active defense depends on the ability of the target to detect incoming missiles with enough time to fire off enough weapons to destroy or decoy them. Timely detection of the incoming

missiles will depend on the sensor network that the defending platform has at its disposal. Airborne radars, air patrols, and good electro optical/infrared systems will help in the detection of incoming weapons. Airborne ESM is especially important in quickly detecting active missile seekers.

Once an attack is detected, the defending platform will automatically take measures to defend itself. The first step will be to activate the various ECM systems that are on board. Effectiveness of the ECM is a function of the difference in technology levels between the missile and the defending ship and the time window that starts with the activation of the incoming missile activating its radar (based on the PAP) and ends with the impact of the missile (on the ship or the water).

The next step will be SAMs. The time window comes into play. It starts with the detection of the inbound missile and ends with either the destruction of the missile or its impact. The length of the window is a function of the incoming missile's speed, the Rate-Of-Fire (ROF) of the launcher, and the speed of the outbound SAM.

22.5. RESOLUTION

Combat resolution can be more properly defined as the moment of impact and is expressed as a probabilistic value "Pk".

There are a number of factors that are calculated at this point. The results are listed in the optional *AA Log* (which can be activated in the H3 Launcher). An example file will look like this, for an engagement between two Harpoon IC missiles and an Israeli *Reshev* PTG.

```
----------------------
EXAMPLE 1
----------------------
Weapon Harpoon IC is resolving its attack against
Yafo
Attacking a surface target with base pH: 80%
++++++++
+ Starting Point Defense Calculations.
+ Defending Unit is Yafo.
+ Target weapon is Harpoon IC.
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is GPMG.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
++ Bang! (Shooting Gun at it)
++ 1 X GPMG fired in Point Defense!
++ 0 GPMG left.
+ Number Shot Down by Mount was 0
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is GPMG.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
+ Number Shot Down by Mount was 0 - (This
mount didn't fire due to its database mount arc
restrictions (eg.starboard mount versus a port
attack.)
```

++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is GPMG.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
++ Bang! (Shooting Gun at it)
++ 1 X GPMG fired in Point Defense!
++ 0 GPMG left.
+ Number Shot Down by Mount was 0
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is GPMG.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
+ Number Shot Down by Mount was 0 - (This mount didn't fire due to its database mount arc restrictions (eg.starboard mount versus a port attack.)
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is 20mm Mk15 Block 0.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
++ Bang! (Shooting Gun at it)
++ BOOM! (Got it :)
++ 1 X 20mm Mk15 Block 0 fired in Point Defense!
++ 0 20mm Mk15 Block 0 left.
+ Number Shot Down by Mount was 1
+++++++++
Shot down by point defense.

The Yafo has survived the first incoming Harpoon, the second may be a different matter:
Weapon Harpoon IC is resolving its attack against Yafo
Attacking a surface target with base pH: 80%
++++++++
+ Starting Point Defense Calculations.
+ Defending Unit is Yafo.
+ Target weapon is Harpoon IC.
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is GPMG.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
+ Number Shot Down by Mount was 0
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is GPMG.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
+ Number Shot Down by Mount was 0
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is 76mm/62 - DP.
++ Incoming Weapon is Harpoon IC.

```
++ Firing Unit is Yafo.
+ Number Shot Down by Mount was 0 — The above mounts
were either reloading or masked by arc restrictions.
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is 20mm Oerlikon.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
++ Bang! (Shooting Gun at it)
++ 1 X 20mm Oerlikon fired in Point Defense!
++ 0 20mm Oerlikon left.
+ Number Shot Down by Mount was 0
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is 20mm Oerlikon.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
+ Number Shot Down by Mount was 0
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is 2nd Gen - Chaff (I/J/K Bands).
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
++ Bamf! (Missile Decoy Effective in EM Mode at -10%
) : 2nd Gen - Chaff (I/J/K Bands)
++ 4 2nd Gen - Chaff (I/J/K Bands) left.
+ Number Shot Down by Mount was 0
++ Resolving Weapon Fire for Point Defense.
++ Defending Weapon is 2nd Gen - IR Flare Decoy.
++ Incoming Weapon is Harpoon IC.
++ Firing Unit is Yafo.
++ Pop! (Missile Decoy Effective in IR Mode at -10% )
: 2nd Gen - IR Flare Decoy
++ 4 2nd Gen - IR Flare Decoy left.
+ Number Shot Down by Mount was 0
+++++++++
Total countermeasures: -10% - (2nd Generation Chaff
attack penalty)
Final pH: 70%
Roll: 56 (Hit)
----------------------
----------------------
```

As you can see, decoys such as SRBOC are used in a point defense mode, along with gunfire resolution that is applicable. It can be noted in the first example above that the incoming target was shot down and the defending ship did not require the expenditure of decoys.

23. H3ANW SCENARIO EDITOR

23.1. TUTORIAL I: SCENARIO DESIGN

Men of sense often learn from their enemies. It is from their foes, not their friends, that cities learn the lesson of building high walls and ships of war. -Aristophanes

Creating a challenging scenario requires much forethought and planning. Here we will cover the process followed by the scenario designers at Three-Sixty, the creators of the original Harpoon 2 scenarios that have now been converted to use the HUD3 Database (one of the official databases).. It is not the only method, but it has a proven track record. You need not run the Scenario Editor to complete this tutorial.

The scenario creation process is composed of three main parts:

1. Crafting the Conflict: the foundation of a scenario; laying in a storyline.
2. Research: finding detailed information based on the preliminary conflict sketch.
3. Implementation: scenario creation by using the scenario editor to construct the foundation for the storyline.

23.1.1. CRAFTING THE CONFLICT

Designing your storyline is the most important step in the creation process. A good design will save time when implementing the scenario. A poor one can result in hours of wasted effort. Since Harpoon 3 simulates modern naval warfare, it is safe to assume that there will be some manner of conflict in your story. The better crafted the conflict, the more enjoyable the scenario. To help stimulate your imagination, ask yourself several questions:

- » WHO are the groups in conflict?
- » WHY are they in conflict?
- » WHERE does the conflict take place?
- » HOW will this conflict be resolved?
- » Who are the groups in conflict?

Let's say we have three groups. Since we would rather keep this tutorial scenario as neutral as possible, let's call them Alpha, Bravo and Charlie.

Why are they in conflict?

Normally you would draw upon your knowledge of historical conflicts involving these groups to answer this question. Doing so will lend realism to your scenarios and help suspend disbelief better than a story with no background.

For this lesson, there is no past conflict or political situation to draw upon, so it must all be created from scratch. Let's say that two of the groups are in conflict: Alpha and Bravo. Perhaps side Charlie can be allied with side Alpha, but neutral toward side Bravo.

Where does the conflict take place?

Since the sides we created have no historical geographical locale, we'll just choose one, Iceland. Let's give Bravo controls of Iceland, and Alpha plans to invade.

How will this conflict be resolved?

The conflict here is quite straightforward. Alpha invades Bravo. We'll say that to win, Alpha must successfully land troops somewhere on Iceland. For Bravo to win, they must successfully hold off this invasion by destroying some of Alpha's key units.

Now that some of the key questions are answered, actual research can begin.

Research

Two items must be researched before they can be implemented in a scenario:

Geography and Game Tempo

Order of Battle

23.1.2. GEOGRAPHY AND GAME TEMPO

Once you decide where the scenario will take place, you must decide how large to make the "battlefield", and what sort of units to include. The size of the theatre of operations and the number of units are determining factors for the tempo of a scenario.

Quick Tempo: 5x5 degrees, littoral surface conflict

Average Tempo: 10x10 degrees, short range air assets, full surface groups

Slow Tempo: 20x20 degrees, long range air assets, carrier groups

Very Slow: 50x50 degrees, multiple air bases, carrier groups with long transit times

We want this scenario to have an average tempo, but we want a larger map, so we will make it a 10 x 25 degree map, mitigating the size by including a small number of units.

Order of Battle

Now that we know the groups involved in the conflict, the geographical region, and the desired tempo of the game, we should now decide which units to place in the scenario for each of the participants. This tutorial can be executed with any ODB derived database, but was updated for ANWDB in particular.

Side Bravo

Base

C3CM Bunker (NATO)

C3I (Comm Center)

Radar Site (Western)

Runway (8 k ft)

Hangar (large x4)

8 F15-E Strike Eagle

8 F15-C Eagle

Side Alpha

Task Force

PK Moskva

6 Yak-38 Forger A

6 Yak-38 Forger A

3 Ka-29 Helix AEW

3 Ka-25 Hormone A

BKR Slava

1 Ka-27 Helix A

BDK Alligator

ARKR Admiral Ushakov

3 Ka-25 Hormone A

Side Charlie

Commercial Trawler Lady Erika

Commercial Trawler Lady Esmerelda

Summary

Once the groups, reason for the conflict, geographical region, and game tempo have been decided, it is time to actually start implementing the design of the scenario. First step: Map Generation.

23.2. TUTORIAL II: MAP GENERATION

In this tutorial you will create the map mentioned in Tutorial I: Scenario Design. You will also learn how to set the starting date, time and duration for your scenario. Unlike the first tutorial, you will need to have the Scenario Editor running to demonstrate this lesson. Once Scenedit has been run, you should see the Load BattleSet Dialog box. Click on the Load File button, and then click on Cancel.

23.2.1. NEW SCENARIO

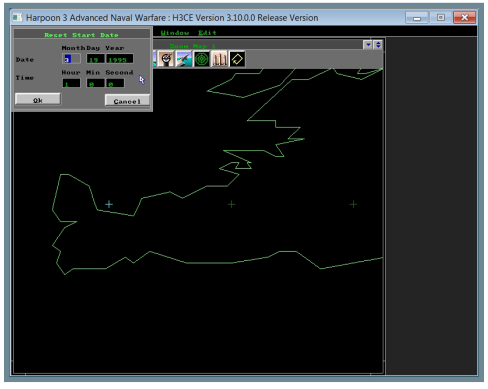
Open the **Edit** pull-down menu. Here are all of the Scenario Editor options and tools can be found. Select **New Scenario**. This will bring up the **Scenario Data Selection** dialog box. From here you will set the scenario start date and time, the duration, map latitude and longitude, and the type of map projection. First let's set the start date and time.

23.2.2. SCENARIO DATA SELECTION DIALOG BOX

23.2.2.1. START DATE

The dd/mm/yyyy represents the starting day, month, and year. Enter **19, 3** and **1995** for the starting date. This means your scenario will begin on March 19. 1995. The starting date is important because it can determine which units are available for placement in the scenario. There are two options set in the H3 Launcher (and stored in the Harpoon3.ini file) called ClassRestrictionByCountry and ClassRestrictionByTime. If you have these options selected you will not be able to place a unit that entered service after the scenario starting date.

NOTE: You cannot create a scenario with a start date before 1900 and a scenario can not run past 2154. The coastline and country border data that Harpoon 3 uses dates from 1976 through 1994.



23.2.2.2. START TIME

Time in Harpoon 3 is measured according to ZULU, or Greenwich Mean Time. This is the time as measured in Greenwich, England. To set the starting time accurately in your scenarios you should take the time difference into account.

First, decide on the geographical region in which you want your scenario to take place. In this case, we decided the scenario should take place around Iceland.

Second, decide the time of day you want the scenario to begin. Let's say we want this scenario to start at 1:00 am. Calculate the ZULU time by adding or subtracting the time difference between your scenario's region and the time in Greenwich, England. Consulting the accompanying map, we can see that Iceland is in the same time zone as Greenwich, so no calculation is necessary for the start time in this tutorial.

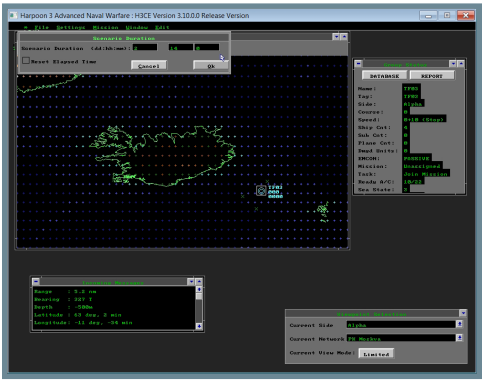
The hh/mm/ss in the Scenario Data Selection dialog box represents the starting hour, minute and second. Enter **1 – 0 – 0** for 1 A.M or 0100 Hrs Zulu time.

However, to demonstrate the time difference, let's say we have a scenario based in the Mediterranean. Consulting the World Time Zones map, we can see that the Mediterranean Sea is in the -1 column. This indicates that we should subtract one hour from the desired time of day to get the ZULU time.

Finally, if you are considering daylight in your scenario time selection, then remember that at extreme northern and southern latitudes, the number of hours of daylight can vary from a very few to all but a few hours each day depending on the time of year.

23.2.3. SCENARIO DURATION

Now we will set the scenario duration. The duration of a scenario must be carefully considered. The scenario must be long enough to accommodate the movement of units and the completion of specific scenario objectives, but short enough to keep the interest of the player. The scenario duration can be altered later in the scenario design process, but it cannot be set to zero when the map is created.



The dd/hh/mm represents the scenario's duration in days, hours and minutes. Enter **1,0,0** now for one day. This will be changed later in the scenario creation process.

23.2.4. MAP BOUNDARIES/ GEOGRAPHICAL REGION

It is important to carefully consider map boundaries before setting them. As mentioned in the first tutorial, the size of the map dramatically affects game tempo. Values for the map boundaries are measured in latitude and longitude. As mentioned before, we want this scenario to take place around Iceland and we want it to have an average tempo, so after consulting the map we will use these values:

- » **Northern Latitude:** 70
- » **Southern Latitude:** 60
- » **Eastern Longitude:** -5
- » **Western Longitude:** -30

Using these Lat/Long values will generate a map 10x25 degrees centered on Iceland.

23.2.5. MAP PROJECTIONS

Once the location of the scenario has been decided and you have entered the dimensions in latitude and longitude, you must decide which map projection to use. A map projection is a means of representing the lines of latitude and longitude of the globe on a flat sheet of paper, or in the case of Harpoon 3, a two-dimensional tactical display. There are two types of map projections available in Harpoon 3: **Miller Cylindrical** and **Cassini Polar**.

23.2.5.1. MILLER'S CYLINDRICAL PROJECTION

This map projection is created by wrapping a cylinder around the globe, touching at the equator. The globe is projected onto the cylinder, then the cylinder is 'cut' vertically at either the prime meridian in the case of the (0,0) projection or the 180 degree line (0,180). Because this projection is most accurate near the equator, the Miller Cylindrical projection should be used when the map area does not include a region around one of the poles.

23.2.5.2. CASSINI POLAR PROJECTION

This map projection is used primarily to display the North and South Polar Regions in Harpoon 3. Instead of having straight meridians and parallels, the Cassini projection has complex curves for all but the Equator, central meridian, and each meridian 90 degrees away from the central meridian. Because this projection distorts the topography as distance from the center of the map increases, it should only be used when your scenario takes place very close to one the poles.

Because we have chosen an area around Iceland that does not include much of the North Polar region, click on the radio button next to the **Miller Cylindrical Projection** (0,0).

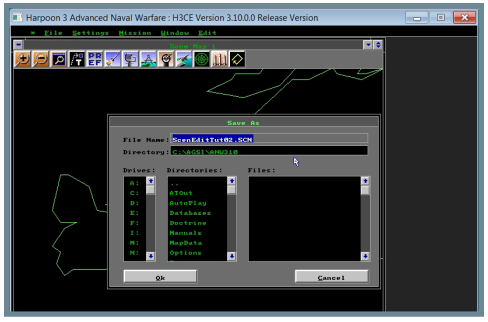
23.2.6. WEATHER

The weather model in Harpoon 3 is a 'global' model. Weather conditions are affected by region, time of year and time of day. In addition, the size of the map plays a major role in the initial 'seeding' of the weather patterns. Larger maps tend to result in more variation of weather patterns than smaller ones. Once the patterns are seeded, they change and develop as the scenario is played, thus the scenario duration also influences the evolution of weather patterns. A larger map is more likely to have intense and varied weather patterns, but it will also result in a decrease in game play speed. For this scenario, **leave the Computer Generated Weather checkbox marked.**

eNow that all of the data for the creation of the map has been entered, click on **OK**. You will see a window with the label "Untitled" appear. It should be centered on Iceland, having the dimensions you entered in the Scenario Data Selection dialog box. Click on the **Prefs button** (F8) and change the name of the window to "**TACTICAL OVERVIEW**".

23.2.7. SAVING

It is important to save after each significant step in the scenario creation process. Use Save As from under the File pull-down menu to save this scenario as "**MAP.SCN".**



23.2.8. SCENARIO DESCRIPTION

After saving the scenario you will be prompted to enter a scenario name and description. Enter the scenario name as "**ICELAND"**. And type "**ICELAND TUTORIAL SCENARIO"** for the scenario description. This will be changed later in the scenario creation process.

23.2.9. SUMMARY

Map generation is one of the most crucial elements in designing an effective scenario. The size of the map affects game speed and weather patterns, and the start date and time can affect the selection of uAnits available for placement in the scenario. Now that the map has been generated and we have an arena of combat, let's add the groups in conflict.

23.3. TUTORIAL III: CREATING SIDES

In H3ANW , the number of possible sides is large, and each can be played by a human player. However, in the Scenario Design tutorial, it was decided to have only three groups: Alpha, Bravo, and Charlie. This tutorial will demonstrate how to make these group's sides and set the posture for each.

Select **Sides** from the **Edit** pull-down menu. You should see the **Side Maintenance** dialog box.

23.3.1. ADDING THE SIDES

23.3.1.1. ADDING SIDE ALPHA

Click on the **Add Side** button. The New Side dialog box appears then type in "**Alpha**" and click on **OK**. You should now see Alpha in both halves of the Side Maintenance window.

23.3.1.2. ADDING SIDE BRAVO

Click on the **Add Side** button and type "**Bravo**" in the New Side dialog box. Then click on **OK**. You should now see Bravo in both halves of the Side Maintenance dialog box.

23.3.1.3. ADDING SIDE CHARLIE

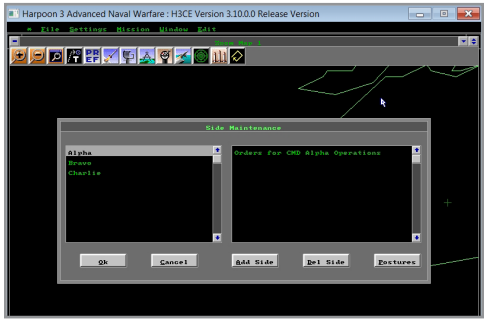
Repeat the steps used to create Alpha and Bravo to create side "**Charlie**".

23.3.1.4. DELETING A SIDE

To delete a side, select it in the left panel of the Side Maintenance dialog box and click on the **Delete Side** button.

23.3.2. ORDERS

Select **Alpha** in the left panel. You should notice that side Alpha now appears on the right as well. The right panel is where the orders for the selected side are displayed. Notice that when you move your mouse pointer over the right panel it seems to disappear. Actually the pointer is being replaced with a text insert pointer that matches the background in color. Click on **Alpha** in the right panel. Notice the cursor is blinking. The right panel is a text editor used to enter orders for the selected side. Backspace over Alpha and type in the following: "**ORDERS FOR CMDR ALPHA OPERATIONS"**



Click on **Bravo** in left panel. Then click in the right panel. Backspace over Bravo and type in the following: **"ORDERS FOR CMDR BRAVO OPERATIONS"**

These will serve as placeholders for the orders text. Later, once the scenario is nearly complete, you will enter the actual orders. Don't enter an orders placeholder for side Charlie.

23.3.3. POSTURES

There are three postures in H3ANW : *friendly*, *neutral*, and *hostile*. A posture defines a side's attitude toward the other sides in a scenario. In more practical terms, a posture determines whether units attack a contact (*hostile*), volunteer sensor information (*friendly*), or neither (*neutral*). Unless changed in the **Settings** pull-down menu, friendly units are displayed as blue, neutral as green, and hostile as red.

A side should never hold a friendly posture toward a side that considers it hostile. This would result in the friendly side transmitting the location of all of its units and sensor information, while the hostile side immediately used this information to destroy those units.

23.3.3.1. SIDE ALPHA

Select Alpha from the left panel of the Side Maintenance dialog box. Click on the **Postures** button. You should now see the Change Posture for Alpha dialog box. The display list will show all the sides other than Alpha. It is important to understand that with Alpha selected, we are defining Alpha's stance toward the other sides.

23.3.3.2. CHANGE POSTURE DIALOG BOX

Notice that Bravo is highlighted. The **Change Posture** dialog box automatically selects the side first on the list, and chooses Neutral as the default posture. We want to make Alpha hold a hostile posture towards **Bravo**, so click on the **radio button** next to **Hostile**, then click on **Bravo**. We want to make Alpha hold a neutral posture toward side **Charlie**. Since the default posture is neutral, we can just leave it as it is. Then click on **OK**.

23.3.4. POSTURES FOR SIDES ALPHA, BRAVO AND CHARLIE

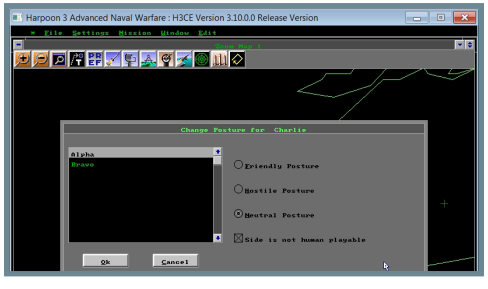
23.3.4.1. SIDE BRAVO

Now we will set the posture for side Bravo. Click on **Bravo** in the **Side Maintenance dialog box** and then select the **Postures button**. First, let's make side Bravo hostile toward Alpha. Click on the **radio button** next to **Hostile**, then click on **Alpha**. We want Bravo to be friendly to **Charlie**, so click on **Charlie** in the left panel dialog box. Select the **Friendly** posture and then click on Charlie again. Click on **OK** to exit.

23.3.4.2. SIDE CHARLIE

Select Charlie in the **Side Maintenance** dialog box. Remember that we have not typed in a place holder for Charlie's orders. This is because we want to leave Charlie a computer-only controlled side. Click on the postucres button. Mark the **checkbox** at the bottom of the Change Postures dialog box labeled "**Side is not Human Playable**." Now we will set the posture for side Charlie.

We want to make Charlie friendly toward Bravo and Neutral toward Alpha, so click on Bravo, then click on the **Friendly posture radio button** and **click on Bravo again**. Since the default is neutral, we will leave Alpha as it is. Click on **OK** to exit.



23.3.5. SUMMARY

The use of Postures adds a 'political' element to the conflict in a H3ANW scenario. With a little research into the history of the sides involved, the influence of their political attitudes on the conflict can be effectively modeled. Next, we will actually place units in the scenario. First, save the scenario as SIDES.SCN.

23.4. TUTORIAL IV: ADDING UNITS

In this tutorial you will add units for sides Alpha, Bravo, and Charlie. Refer to the Order of Battle we created in Tutorial I: Scenario Design.

For the purpose of this tutorial, make sure that you have both the **Class Restriction By Country** and **Class Restriction By Time** options in the Harpoon 3 Launcher (**Game Preferences Tab**, **Scenario Editor Options**) toggled **OFF**. Having either of these options ON will result in erroneous results in this tutorial.

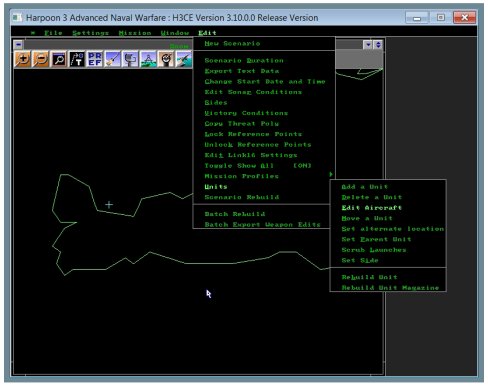
23.4.1. SIDE BRAVO

Let's add units for side Bravo first. Click on **Sides** from the **Edit pull-down menu** and choose side **Bravo**, then click on **OK**. Then **create a zoom window** centered on the southwestern most peninsula of Iceland, and **zoom** in until the peninsula fills most of the window.

23.4.1.1. ADD A UNIT

Let's put the Bravo base in this area. Go to the **Edit** pull-down menu. Select **Units** then **Add a Unit.** Notice that the cursor has changed to a hand and pointed finger. The Scenario Editor is prompting you to locate the unit.

Click near the end of the peninsula, midway between the North and South shores. You should see the **Select Unit Type** dialog box. The **Select Unit Type** dialog box is terrain sensitive. The available unit types are:



Unit Types (Land)

Aircraft - places an airborne aircraft

Facility - places a facility (includes: radar-sites, missile-sites, communications, etc.)

Installation - a prefabricated base

Unit Types (Sea)

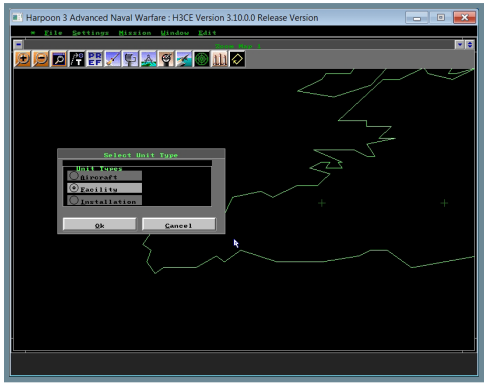
Aircraft - places an airborne aircraft

Ship - places a surface ship

Submarine - places a submarine

NOTE: With Num-Lock OFF, you can also use the keyboard hotkeys, using the 0 (Ins) key on the numeric keypad to add a unit, and the "." (Del) key to remove a unit. Also, the CTRL+A Command will drop the same unit as selected by the user.

A base is actually a group of facilities, so click on the **radio button** next to **Facility**. Press **OK**. You should see the **Facility Select** dialog box appear. Scroll down until you see the C3CM Bunker. Select it and then click on **OK**. An icon for the bunker should appear on the map.



Notice that the bunker has the flagship designator. The first unit placed for a side is the flagship (keep in mind that if you play at the higher reality levels, you can determine which unit is at the center of your Comm network). Using the **Numeric Keypad "*"**, turn on the display for data blocks. Next, toggle **Range and Bearing ON** from the Settings pull-down menu (or **Numeric Keypad "5"**), then adjust the **zoom level** on the map until the distance from the center of the facility icon to the right edge of the C3CM text is one nautical mile. *This is a ranging convention used for the purposes of this tutorial to help place the other facilities; it is not required when making later scenarios.*

Drag-select a blank spot on the map to de-select the bunker, then turn off the display for data blocks.

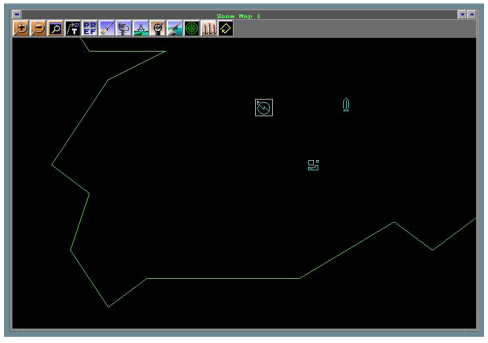
Select **Add** a unit from the **Edit pull-down menu**. Click next to the icon of the bunker about where the data block should be, and select **Facility** as the unit type. Click on **OK**, then find and select the **C3I** (Comm Center). Press **OK.**

You should see another facility icon appear, and if you have data links marked in the Map Preferences dialog box, you will see the link between the Comm center and the Bunker.

23.4.2. A NOTE ABOUT ICONS

Facilities are displayed in three ways: a weapons icon, a communications icon, or a custom icon. If a facility has a weapon, it displays as a weapons icon. If it has communications equipment, it will be displayed as a communications icon. If it has both, it will be displayed as a weapons icon, and it if has neither a weapon or communications equipment, it will be displayed as a custom icon.

- » Weapon Facility Icon (upper right in picture)
- » Communications Facility Icon (upper left in picture)
- » Custom Facility Icon (building lower center in picture)



Next we will add the radar site for Bravo's base. Select **Add a Unit**, then click on the south side of the bunker about the same distance as the Comm center. Choose **Facility** and then select **Radar Site (Generic West Bloc).** Notice it appears as a weapon rather than a communications icon.

We want Bravo's base to have air capabilities, so we will need to place a runway and a hangar. Follow the same procedure as before, placing the **Runway (Very Large Aircraft x4)** to the west and the **Hangar (Large Aircraft x 4)** to the north. A runway and hangar are the two required elements of an air base.

To help protect the air base, place the **SAM (Patriot battery)** to the southwest between the runway and the radar site.

As mentioned before, a base is a type of group. Its sea-going parallel is the task force. Like the task force, we need to 'form' the base by grouping its components, in this case the facilities. Do not group them quite yet as we need to explain a significant difference between a task force and a base, namely the detection level.

23.4.3. AUTO DETECT

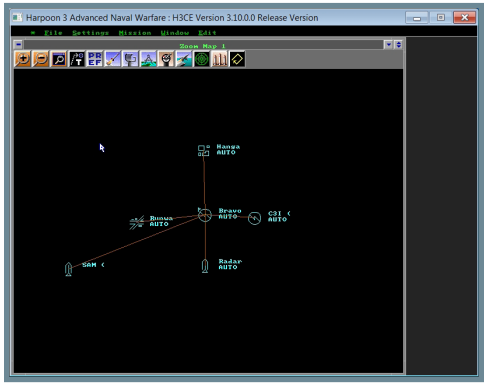
One important consideration when forming a base is the intended level of detection. Facilities that are manually joined to the base are auto detected by the other sides when the scenario starts, so for the sake of stealth, it is sometimes good strategy to leave certain elements unattached. Let's say we want the SAM site (Patriot battery) to be hidden when the scenario begins. Click on the SAM site and then hit the Home key (not the 7 on the numeric keypad). Notice the word AUTO goes away. Now, as long as we do not make it a part of the base, it will not be auto-detected by the other sides when the scenario begins. To make sure we don't include it as part of the base, let's move it to the west.

23.4.4. MOVE A UNIT

Click on the SAM site, and then select **Move a Unit** from **Units under the Edit pull-down menu**. Notice the pointer changes to a hand. **Click** the hand pointer on a spot several miles to the west, so that it is no longer directly under the runway icon. You should see it appear where you clicked. You can also move an individual ship, or even a group of ships; however, you cannot move bases. Now that the SAM site is out of the way, let's form the base.

23.4.5. FORMING A BASE

Drag-select all of the **facilities** with the exception of the SAM site, once all are selected, press the **'G'** hotkey to group them. Toggle into group view mode, and you should see that the facility icons have been replaced by a single base icon. Bravo's base is now formed, and the SAM site will be hidden from the sensors of the other sides until detected through the course of the game.

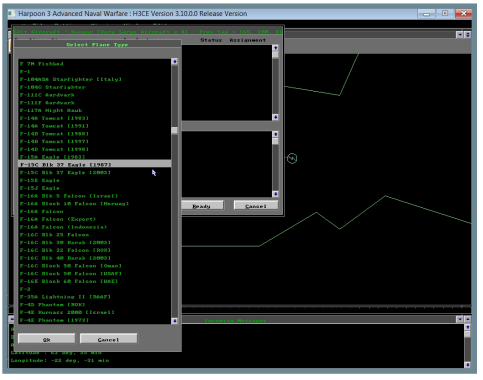


23.4.6. EDIT AIRCRAFT

Now we will add the rest of Bravo's units. There are two ways to place Aircraft: Add a Unit, and Edit Aircraft. Because Add a Unit is used to place airborne aircraft, let's use the Edit Aircraft option. Zoom in on the base and switch to unit mode.

Click on the **Hangar** and then select **Units** and **Edit Aircraft** from the **Edit pull-down menu**. Keep in mind this only works if you have created a Base that includes the hangar and an appropriate runway.

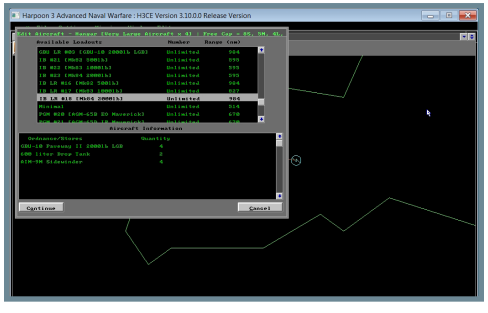
23.4.7. EDIT AIRCRAFT DIALOG BOX



Click on the Add button. This should bring up the **Select Plane Type** dialog box. You will see a list of the aircraft available for placement in the scenario. This list contains only those units that will fit in the selected platform (this applies more to ships than bases). Find and select the **F15C Blk 37 Eagle (1987)**, then click on **OK**. A **box** will appear asking you how many F15C's to add. This default is determined by the capacity of the "holding" platform. In this case, the hangar's capacity is 10. Change this number to **5,** and click on **OK**.

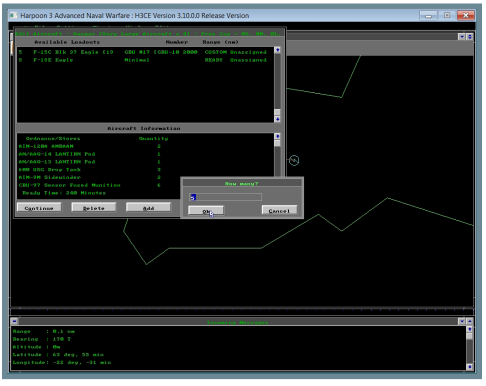
23.4.8. READY AIRCRAFT

Now that we have placed the F15's, let's ready them. Click on the **Ready button**, then select one of the ATA loadouts, either **#18** or **#19**. When prompted with "How Many?", click on **OK** to loadout all eight aircraft. You should see a dialog box asking 'Minutes to Ready?' Notice that the default time is -1 minutes, which means that the time used when the game starts will be determined by the database (and thus loadout) in use for the scenario.



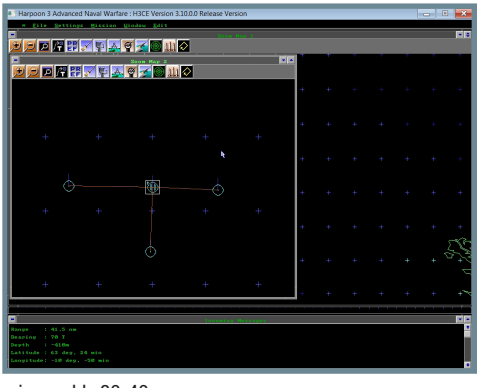
Ready Time is where the Edit Aircraft dialog box differs considerably from using the **Add a Unit** option. This is the amount of time the selected aircraft will require to ready the **first loadout** once the scenario begins.

For example, if you were to type “60” in the Minutes To Ready? Dialog box, the 8 F15’s would take one hour to ready once the scenario begins. However, following this initial time period, each future loadout in the scenario will return to the standard 30 minutes. This is used by Scenario designers to improve play balance and realism.Follow the same procedure to place 5 more F15E’s in this hangar. This time, ready them with the iron bomb (IB) or PGM loadout of your choice.



Now we are ready to add units for side Alpha, so save the scenario as **ADDALPHA.SCN**.

23.4.9. ADDING UNITS FOR SIDE ALPHA



Zoom out so that you can see Iceland and the islands to the southeast. **Choose Sides** from the **Edit pull-down menu**, and select side **Alpha**. Click on OK and from the Edit pull-down menu, select Add a Unit from Units. Click the hand cursor at a point 2/3 of the way between Iceland and the islands to the southeast,

Select the **radio button** next to **Ship** and click on **OK** and scroll down and select the **PK Moskva**, which is a VTOL carrier. Create a zoom window centered on the Moskva, and adjust the zoom level until the zoom area or window is roughly 30-40 nm across.

Add a ship 10 nm to the east of the Moskva and make it the cruiser **Project 1164 Atlant (Slava) RK** . Next add a unit 10 nm to the south of the Moskva and make it the landing ship **Project**

1171 Tapir (Alligator) BDK. Finally, add a ship 10 nm to the west of the Moskva and make it the **Project 1144 Admiral Ushakov**, a nuclear-powered battle cruiser.

23.4.9.1. CREATING A SURFACE GROUP

The procedure for creating a surface group is similar to the one we used earlier in the tutorial to create a base. **Drag-select** all of the vessels and hit the "**G**" hotkey to group them together, then switch to group view mode **(Keypad 9 hotkey)** to verify the group has been created.

To add aircraft for Side Alpha, toggle into unit view mode **(Keypad 9 hotkey)**, and **select the Moskva**. Select Edit Aircraft from Units in the Edit pull-down menu. Click on the Add button. Select the Yak-38 Forger A. Ignore the selected default number of units, change it to **6**. Once the six Yak-38's appear in the upper panel of the **Edit Aircraft** dialog box, click on the **Ready** button. We want this group of Forgers to have an anti-air loadout, find and select the **ATA LR#02 (AA-8 Aphid).** Ready all 6. Once back at the Edit Aircraft dialog box, add 6 more **Yak-38 Forger's,** but this time ready them with the **IB #177 FAB 250 (250kg bombs)** loadout.. Finally, place 3 **Ka-29RLD Helix D AEW's** and 3 **KA 25 Hormone A's** on the Moskva. We want to leave these two groups of aircraft with a minimal loadout, so once they appear in the upper panel of the Edit Aircraft dialog box, click on **Continue.**

To finish adding air units for side Alpha, select the **Slava**, and add one **KA 27FL Helix A**. Then select the **Admiral Ushakov** and add 3 **KA 25 Hormone A's**. Leave both of these groups with minimal loadouts.

Switch to group view mode **(Keypad 9 hotkey)**, and **select the task force**. You may now edit your formation as you see fit. Reference the H3ANW Basic manual for details. Once done, save the scenario as ADDCHARL.SCN.

23.4.10. ADDING UNITS FOR SIDE CHARLIE

Select side Charlie from the **Sides** option in the Edit pull-down menu; then click on **OK**. At this point make sure that **Show All** (if you lost it go to the **Window menu** and select it from the bottom of the menu) is currently on.

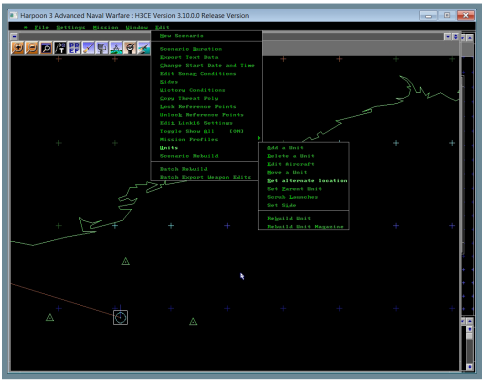
Make sure the water depth checkbox is marked in the map preferences, then **add a unit**, centering it between the base and Alpha's task force. Choose **Ship** as the unit type, and make it a **Merchant Ship Fishing Vessel, rename** the trawler **"Lady Erikia"**. Then add another unit roughly 30 to 60 nm south of the Fishing Vessel. Make this vessel another **Merchant Ship Fishing Vessel**. To avoid confusion, select it and hit the **7** (home) key on the numeric keypad. Make sure the num-lock is off. **Rename** the trawler the "Lady Esmerelda".

23.4.10.1. ALTERNATE STARTING LOCATIONS

One of the useful features in the Scenario Editor for increasing scenario "replayability" is the Alternate Starting Location option. When the scenario begins, the computer randomly chooses one of these locations to place the unit. Let's do this with side Charlie first. Add **reference points (**using the **Mission menu** or the **"0"** key on your keypad) in a triangular pattern around the Lady Erika.

23.4.10.2. SETTING ALTERNATE STARTING LOCATIONS

Drag-select the three reference points, and the Lady Erika. Then **Set Alternate Location** from **Units** in the **Edit pull-down menu**. Now when the scenario is played in a non-Scenario Editor version of H3ANW , the Lady Erika will randomly start at one of these locations.



Next add three reference points around the Lady Esmerelda. Select these points and the Lady Esmerelda and set them as alternate starting locations.

> **Warning: Do not use any of the reference points laid as starting locations for the Lady Erika. Two units cannot share Alternate Starting Locations.**

Now switch sides to side Alpha and toggle into group view mode. Place three reference points around Alpha's task force, and make them alternate starting locations.

23.4.10.3. REMOVING ALTERNATE STARTING LOCATIONS

You can remove all the starting locations for a side by selecting the Set Alternate Starting Locations option with no reference points selected.

23.4.11. SUMMARY

When placing alternate starting locations, keep in mind the desired tempo of the game. You should switch to unit mode with all range circles displayed, so that you can estimate transit and engagement times. This will be detailed in Tutorial VI: Victory Conditions. Save the scenario as UNITS_IN.SCN.

23.5. TUTORIAL V: MISSIONS

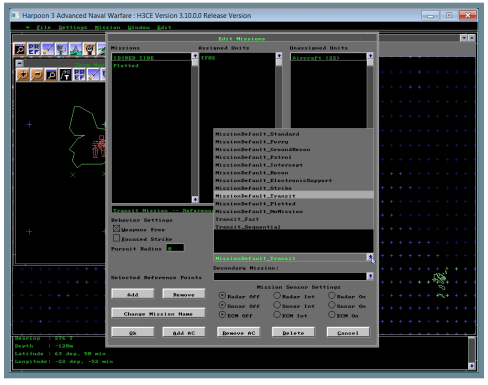
In this tutorial you will create missions for the units. Because mission creation is covered extensively in the H3ANW Basic manual, it will not be detailed here. Only those missions relating to the orders are included.

Make sure you are in group view mode, and then **select side Alpha**. Since Alpha's task force includes an amphibious assault ship, we will say that the Alpha task force needs to reach the region of beach near the Bravo base. **Add three reference points** due south of the Bravo base.

23.5.1. TIME DELAY

Next, create a **Transit mission** with a D**elay Time** of **1** or more minutes. Check the **Edit Now** check box, and click on **OK**. In the **Mission Editor**, select the mission you have just created and assign Alpha's task force. Click on **OK**

Select Alpha's task force. Notice the mission slot in the unit status window no longer reads 'unassigned'.



When creating a scenario, always give transit and area missions a time delay of 1 minute or more. If you do not, then as soon as you create the mission (in the Scenario Editor) the unit navigator will plot a course and this course will be the same every time the scenario is played. Making the mission time delay forces the unit navigator to plot the course when the scenario is played rather than in the editor, and the course will be varied.

23.5.2. ADDITIONAL MISSIONS

Referencing the H3ANW Basic manual where necessary do the following:

- » Place all 12 Yak-38's on a cap station,
- » Create an ASW dipping sonar patrol, and
- » Put one of the Helix AEW on a Support mission.

Next switch to side Charlie and place both trawlers on a transit mission with radar active, the idea being to keep them moving around the area, reporting all sensor information to side Bravo.

Finally, switch to side Bravo and lay in missions to defend the base from air attacks. Remember that when you turn this scenario over to be played, the side not played by a human will only have the orders you give it. Save the scenario as MISSIONS.SCN.

Properly constructing the missions for a side provides the computer player with the majority of its 'intelligence'. The mission behavior is certainly an aspect of the AI, but ultimately it falls on the scenario designer to make each side as challenging to play as possible.

23.6. TUTORIAL VI: VICTORY CONDITIONS

Victory Conditions are one of the most difficult aspects of creating a scenario. As a designer, you must take the conflict resolution ideas from the scenario design phase and make them concrete and feasible within the confines of H3ANW . A rough victory condition was laid out in Tutorial I: Scenario Design: "Alpha invades Bravo." Later, when units and missions were created, the victory condition was refined to:" Alpha must transit a troop transport to the beachhead south of the Bravo base." In this tutorial, this victory condition will be broken down further into specific scenario objectives.

NOTE: Victory conditions are checked every 30 minutes of game time. This is done to ensure that the evaluation of the scenario is accurate and reflects events in progress when you reached your victory condition(s). An example might be that you destroyed all enemy ships but there were still enemy missiles in the air at the time you sunk the last enemy ship. The 30 minute delay allows these missiles to reach the end of their flights where they may potentially affect the outcome of the scenario.

First, select side **Bravo** from the **Side Maintenance** Selection dialog box. Then select **Victory Conditions** from the Edit pull-down menu.

23.6.1. VICTORY CONDITION CONCEPTS

Victory Conditions (VC) are a logic tree. They have branches across (destruction of enemies and protection of own assets) and down levels (composite VC hold and/or conditions). They can be broad (i.e Types) or very specific (named Units). The VC Editor is the scenario designer's interface to crafting VC.

Composite VC

A Composite VC (CVC) is used to collect and combine the VCs that are below it. It in effect, adds an "and" or "or" to the VC dialogue. This feature was introduced in v3.7 and is not compatible with the v3.6.3 program engine.

In practice, the CVC must be on top and then you can use the down level button to create the first level of VCs. At that point you would create the VCs normally. However you can modify the and/or switch inside each level by altering the CVC.

The CVC itself is just a number value that you can change by selecting it and hitting the modify button. The number value indicates which of the and/or switch is in effect. If the CVC value is lower than the number of VCs in the level below that CVC then the "or" switch is in effect. If the CVC value is the same as the number of VCs then the "and" switch is in effect.

Examples:

Top Level Composite = 2

Next Level:

- » VC1 = Protect Station
- » VC2 = Destroy Units
- » VC3 = Protect Units

In this case the VC would be To Protect Station OR Destroy Units OR Protect Units

Top Level Composite = 3

Next Level:

- » VC1 = Protect Station
- » VC2 = Destroy Units
- » VC3 = Protect Units

In this case the VC would be To Protect Station AND Destroy Units AND Protect Units

Top Level Composite = 2

Next Level:

- » VC1 = Protect Station
- » VC2 = Destroy Units
- » VC3 = Composite = 3

Next Level:

- » VC1 = Protect Station
- » VC2 = Destroy Units
- » VC3 = Protect Units

In this case the VCs are To Protect Station AND Destroy Units OR To Protect Station AND Destroy Units AND Protect Units

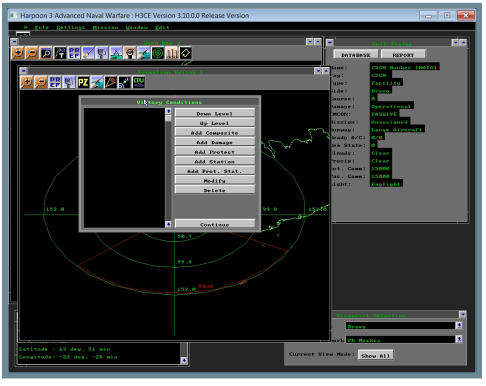
In short:

- » CVC Value = # of VCs = AND
- » CVC Value < # of VCs = OR

23.6.2. THE VICTORY CONDITIONS DIALOG BOX

Because Victory Conditions (VC) can be overwhelming at first, we will walk through the function of each of the buttons before actually using them.

There are nine buttons in the Victory Conditions dialog box:



They are mapped like this:

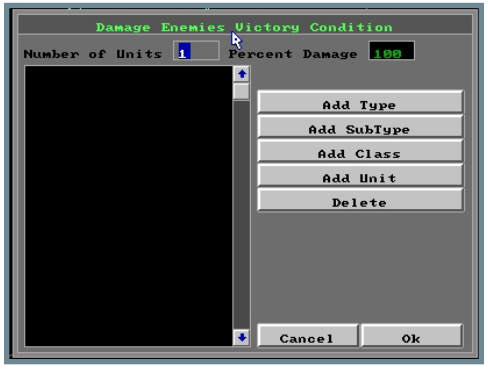
Victory Conditions Buttons	Brings up
Down Level	Only works if there is a VC below your current level. Use this button to scroll 'down' the VC tree..
Up Level	Only works if there is a VC and you aren't at the Top. Use this button to pop 'up' the the VC tree.
Add Composite	Adds a level – see above for implementing this logic
Add Damage	Damage Enemies Victory Condition Dialog
	Add Type, Add SubType, Add Class, Add Unit, Delete, Cancel, OK
Add Protect	Protect Friendlies Victory Condition Dialog
	Add Type, Add SubType, Add Class, Add Unit, Delete, Cancel, OK
Add Station	On Station Victory Condition Dialog
	Relative, Set Poly, Add Type, Add SubType, Add Class, Add Unit, Delete, Cancel, OK
Add Prot. Stat.	Defend Station Victory Condition Dialog

Victory Conditions Buttons	Brings up
	Relative, Set Poly, Add Type, Add SubType, Add Class, Add Unit, Delete, Cancel, OK
Modify	Only works if there is a VC to edit
Delete	Only works on the selected VC
Continue	Exits and saves the VC

We will now explore those buttons that lead to additional Dialog Boxes as explained in the chart above.

23.6.2.1. ADD DAMAGE BUTTON

Clicking on this button will bring up the **Damage Enemies Victory Conditions dialog box**. In this dialog box you can compose a list of specific units or types of units that must be damaged to a given percentage to accomplish a single scenario objective.



Number of Units- The number entered in the **Number of Units** field should not be larger than the number of units in the set shown in the scroll box.

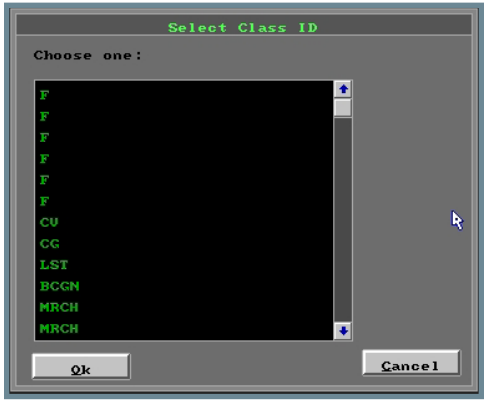
Percent Damage - sets the percent of damage that a unit must sustain in order to for it to count toward the Victory Conditions.

Examples: 4 units at 100% means 4 units must be destroyed. 4 units at 50% means 4 units must be critically damaged. There must be at least 4 units listed

Another possible application of the Percent Damage field is in a scenario where one side has an extreme defensive advantage. You can make the number of units=1, and % damage=5 as a way to allow for a 'political' victory of sorts. If a foe can prove to the populace of the enemy that, it can inflict damage, perhaps the enemy country will back out of a military operation. We will discuss the application of this later in Defend Units.

Add Type Button - If you click on the Add Type button, you will see the **Select Unit Type dialog box**. There are four unit types listed: **Ship, Sub, Aircraft,** and **Facility**. If you choose ship, anything listed as a 'ship' in the H3ANW database will qualify. This should be used to qualify general types instead of individual units in the scenario.

Add Sub-Type Button – If you click on this button, you will see the **Select Class ID dialog box**.



Some values displayed

F – Fighter (any of them)

CV - Carriers (including CVH, CGH, DDH)

CG - Guided Missile Carriers

LST - Transport ships

BCGN - Nuclear Battle Cruisers

MRCH - Merchants

NOTE: As of v3.10 the game still lists duplicate Sub-types – this is only cosmetic.

Add Class Button - Clicking on this brings up the Select SubType dialog box. This is a list similar to the Add Unit dialog box. However, this list includes only those units placed in this scenario.

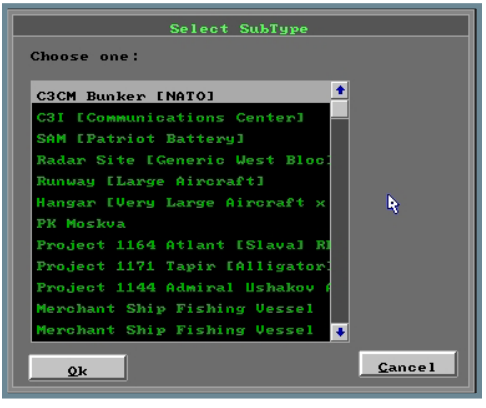
NOTE: It is important when adding units to this scenario objective that you do not add your own units. You will not be able to intentionally damage them and therefore you will not be able to meet the VC.

Add Unit Button - Clicking on this button brings up the Select Unit dialog box. Note the similarity to the Select Class dialog box. However, in this dialog box, when you select an item (such as the C3CM Bunker) you are selecting that individual unit as opposed to selecting a group of those similar units.

Note: If you have multiple C3CM Bunkers (or any other similar unit types) in the scenario, and you have not given them individual names, you will not be unable to differentiate them in the list.

Delete Button- Selecting any item in the list and clicking on the **Delete button** will remove that item from the list.

For the tutorial you should select the **Add Class button** and then choose the **C3CM Bunker**; all the C3CM bunkers in the scenario will be a viable target. As the C3CM Bunker is a facility, the appropriate class for this target (facility) is displayed in the **Damage Enemies Victory Conditions** dialog box.



23.6.2.2. ADD PROTECT BUTTON

If you click on the **Add Protect button**, you will see the **Protect Friendlies Victory Conditions dialog box**. This is nearly identical to the **Add Damage** scenario objective, except that instead of listing enemy units to damage, you are listing your units or those on your communications net (those units you consider friendly). **T**he other significant distinction is that if the number of your units entered in the Number of Units field suffers the amount (%) damage in the Percent Damage field, the scenario objective is NOT reached.

Example: If you enter 4 units with 100% damage as one of the scenario objectives for Protect, then no more than 3 of the listed units can be destroyed.

The Add Protect dialog box has the same buttons as the Add Damage dialog box, and they function in a similar manner.

23.6.2.3. ADD STATION BUTTON

Clicking on this button brings up the **On Station Victory Condition dialog box**. You use this option to define a scenario objective dependent on a certain unit reaching an area (i.e. an amphibious assault craft reaching a beachhead). Most of the buttons on this dialog are the same as the previously covered buttons, the unique fields and buttons are covered here:

At the top of the dialog there are two time entry fields to help define time on station.

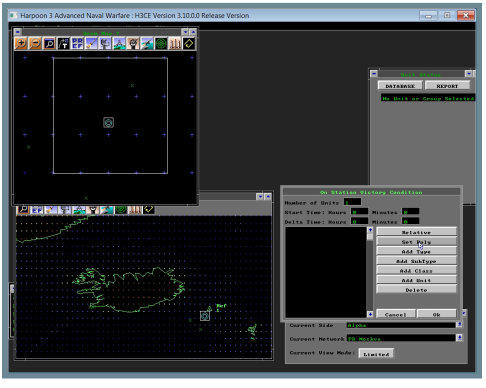
Start Time - The time listed in hours and minutes from the time the scenario begins that the clock starts

Delta Time - The length of time a listed unit must remain in the designated area, or be excluded from it. This is a **single contiguous duration**, not the sum of fragments spent in or out of the designated area.

Relative Button - Click on the **Relative button**. Notice it toggles to **Absolute**. The primary difference between **Relative** and **Absolute** is in the **Starting** Time. If you have **Relative** Selected, the clock starts counting toward the scenario objective when the selected unit reaches the designated area. In an **Absolute** scenario objective, the clock starts at the time listed in the Start Time field.

Set Poly Button - To demonstrate the Set Poly option, move both dialog boxes out of the way for a clear view of the tactical map. Make sure that the right side of the **On Station Victory Conditions dialog box** is visible so that you can reach the **Set Poly button** with your mouse pointer.

Activate the **Tactical Map** and make sure **Water Depth** is **toggled** on in the **Map-Preferences dialog** box. Then **drag-select** an area around the hostile Alpha Task Force, making sure the area is roughly 4 dots by 4 dots. Notice that the area selected disappears. At this time, the fact that the Task Force is selected is inconsequential. You still have the area you drew selected, but it is not visible. **Click** on the **Set Poly button** in the **On Station Victory Condition dialog** box. Notice the area you drag-selected around Alpha's Task Force is highlighted.



This highlighted area is the area the listed units must reach and occupy to complete the scenario objective.

NOTES: The last area you drag-select is the area that will be used when the Set Poly button is clicked. With a Delta Time of 0, the moment a listed unit enters the area defined by Set Poly, the scenario objective is met. Also, remember that we are just doing a walk through now. We will not want this victory condition in our final scenario. This is just an example.

Next, **drag** the **On Station Victory Conditions dialog box** back to the center of the screen, and click on **Cancel**. Then, drag the **Victory Conditions dialog box** back to the center of the screen.

23.6.2.4. ADD PROTECT STATION

Clicking on this button brings up the **Defend Station Victory Condition dialog box**. It is similar to **Add Station**, except in the case of this scenario objective, you must exclude the enemy from entering the designated area. If you have **Relative** toggled on, the clock starts as soon as you start the scenario. If you have **Absolute** toggled on, the clock starts at the time designated in the **Starting Time field**.

The **Delta time** for the **Defend Station dialog box** counts for as long as the listed units are **not in the area**. *If a listed unit enters the area, then the clock resets to zero*, and remember, you must successfully keep units out of the designated area for the full **Delta time** to accomplish this scenario objective. Therefore, if enemy units of the type proscribed enter the area, they must be destroyed with enough game time remaining to make the **Delta time**, or that Victory Condition is lost.

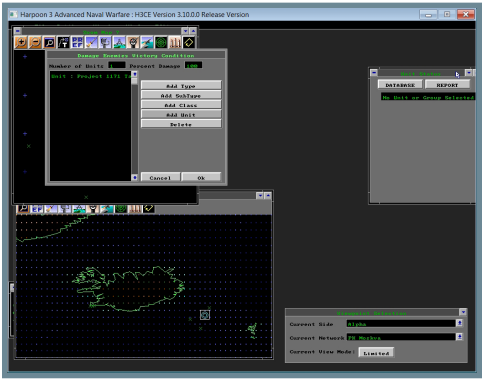
23.6.3. VICTORY CONDITIONS FOR SIDE BRAVO

Now we will actually lay in the Victory Conditions for side Bravo. When considering the Victory Conditions for a side, consider what they can accomplish with their available equipment. Realistically, side Bravo in this scenario cannot sink or exclude the task force because they lack sufficient assets. However, given the tactical situation, they can thwart an amphibious assault by sinking the Alligator or hurting the Slava.

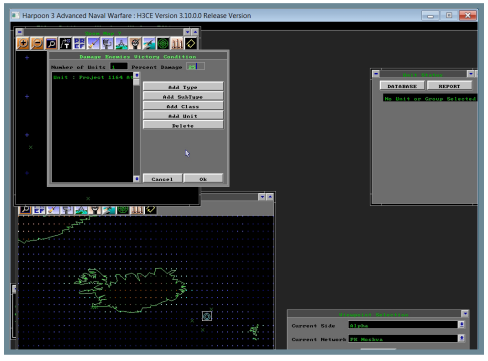
Click on the **Victory Conditions** button from under the **Edit pull-down menu**. Click on the **Add Composite button**, then click on **Composite** on the left hand side, then click **Down Level**.

Select the **Add Damage button**. Leave the **Number of Units** field at "**1**", and the **Percent Damage** field at "**100%".** Now click on the **Add Unit button**.

Select the **Project Tapir 1171 Alligator**. You should see **Unit**: **Project 1171 Tapir Alligator** appear on the list. Then click on **OK**. You should see **Damage/Destroy** in the **Victory Conditions dialog box**.



Click on the **Add Damage button**. Click on the **Add Unit button**, and then select the **Project 1164 Atlant [Slava]**, enter **25** in the **Percent Damage** field, and click on **OK**.

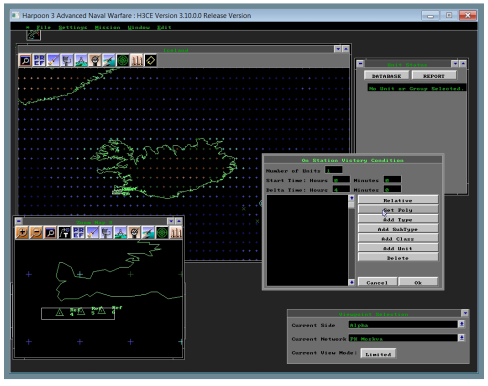


This now gives you a Victory Condition of sinking the Alligator or hurting the Slava (see Composite rules above).

23.6.4. VICTORY CONDITIONS FOR SIDE ALPHA

Switch over to side Alpha. Select **Victory Conditions** from the **Edit pull-down menu**. The primary goal of side Alpha will be to transit the Alligator to the beachhead south of the Bravo base to debark troops and supplies. Click on the **Add Station button**. Move the **Victory Conditions dialog box** down to the bottom of the screen, then drag the **Add Station dialog box** out of the way so that you have a clear view of the Tactical map. Make sure the **Set Poly button** is still in view.

In the current zoom window, zoom in on the Bravo base so you can see the reference points to the south of the base. **Drag-select a box around the three reference points**. **Click on the Set Poly button**. The area you drag-selected will be highlighted.



Move the **On Station Victory Condition dialog box** back to the center of the screen.

Don't worry about the Start Time, but set the **Delta Time to 4 hours**. Make sure that **Relative** is visible on the **Relative/Absolute button**.

This scenario objective means that side Alpha can get the Alligator to the designated area at anytime, so long as it remains in the area for four hours. Remember that if the player of side Alpha gets the Alligator to the area with only three hours remaining in the scenario, it can never meet its objective. The scenario duration must be set with these objectives in mind.

Click on **OK**. **Drag the Victory Conditions dialog box** back to the center of the screen and click on **Continue**.

23.6.5. EDITING SCENARIO DURATION

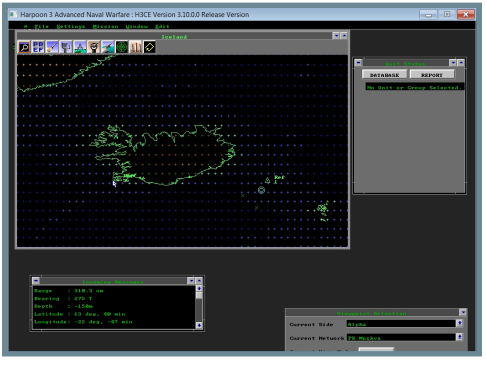
Because many of the Victory Conditions Levels you create will have objectives with time considerations, editing the Scenario Duration is almost always required. The main factor is the travelling speed of the listed units, and the amount of time they must occupy or defend a given area. Because of the repositioning that occurs within a task force as it maintains its formations, the speed of the group can be difficult to ascertain with accuracy. As a general rule, subtract 5 from the group's cruise speed in knots and use it along with the range to calculate the amount of time needed to transit to the destination.

First, zoom out so that you can see the Alpha Task Force, then turn on **Range/Bearing** from the Settings pull-down menu. **Select** Alpha's Task Force, and then change the speed to cruise. Note the group's cruise speed is 10 knots.

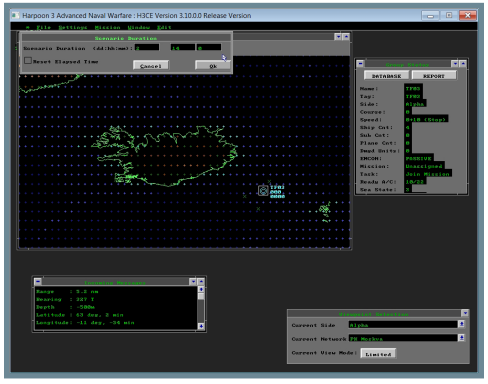
Because alternate starting locations randomize the location of the task force, **click** on the alternate starting location reference point farthest from the designated area. Then **click** once in the area south of Bravo's base. This will give you the longest possible transiting distance for Alpha's task force, and a good range to use for editing the duration.

To find the time you will need to enter as the scenario duration, divide the range by the speed you derived (cruise speed-5). Then **add this to the amount of time required** to accomplish any On Station scenario objectives.

Edit the scenario duration by selecting **Scenario Duration** from the **Edit pull-down menu**, and changing the days/hours/minutes fields.



In this example, the range from **Ref 1** (NE of Alpha) to the farthest target **Ref 4** (tip of the peninsula) is 310nm. Speed is cruise (10)-5 or 5kts. 310nm/5 kts gives a duration of 62 hours, plus the 4 hours that the Alligator has to stay on station or 66 hrs (which is 2 days and about 14 hours).



23.6.6. RESET ELAPSED TIME

In the **Scenario Duration dialog box**, there is a check box labeled, "**Reset Elapsed Time**". Marking this checkbox will reset the start time for all internal applications in the scenario. This feature allows you to play the scenario to monitor movement, or perhaps allow units to engage in combat, then reset the start time so that the scenario begins at the new time.

For example, let's say you have a scenario with a duration of 2 days and 21 hours starting at 2:00 ZULU, and you want to start the scenario with a combat in progress. You could play the scenario for an hour to bring the clock to 3:00 ZULU. If you edit the scenario duration and then mark the Reset Elapsed Time check box, when you click OK, the 2 days and 21 hours will start from the current time (3:00 ZULU). The hour you let the scenario run is not subtracted from the duration or any other time-dependent functions.

NOTE: If you set an absolute start time for one of the Scenario Objectives to 2:30 ZULU, and then let the game run for an hour until 3:00 ZULU, after resetting the elapsed time, there will be a conflict. This is because the Reset Elapsed Time feature does not change the start time for scenario objectives. You cannot have an absolute start time earlier than the actual scenario start time. To prevent this conflict, make sure you set any absolute start times to something later than the modified Scenario Start Time.

23.6.7. SUMMARY

Victory Conditions are one of the more difficult aspects of the H3ANW Scenario Editor, and as such your ability to implement them effectively will grow only with experience. We now have simple scenario objectives based on the naval goals for sides Alpha and Bravo from the scenario design tutorial. As Charlie is a computer controlled player only, it does not use Victory Conditions. Save the scenario as VC_IN.SCN, and move on to the last lesson Tutorial VII: Miscellaneous.

23.7. TUTORIAL VII: MISCELLANEOUS

The H3ANW Scenario Editor includes numerous features to help fine tune or 'tweak' your scenarios. This section of the tutorial is a reference for those features as found, in order, on the Edit menu.

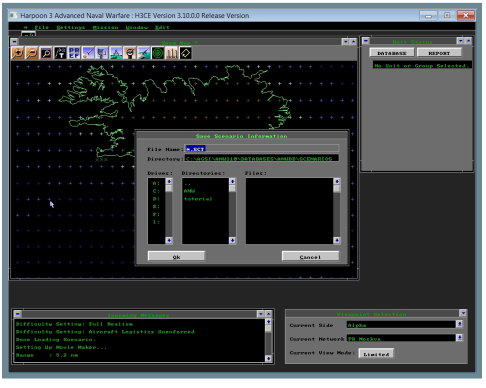
23.7.1. TOGGLE SHOW ALL

On the **Edit Menu**, there is a choice called **Toggle Show_All [off] / [on]** - The Show_All option allows you to see all units on all sides currently in the scenario.

Note: There is a slight idiosyncrasy in the windowing interface program. Although the Scenario Editor starts with Show All toggled ON, in order to actually implement Show All [on], you must turn it off, turn it back on, and then click on the map window to update it.

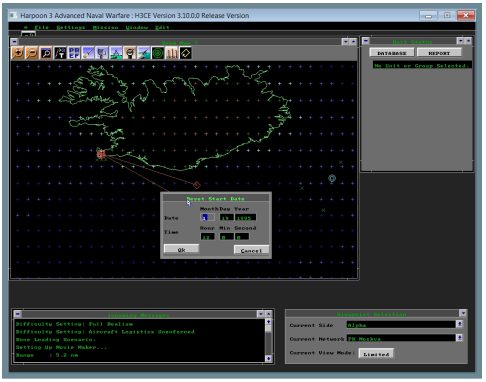
23.7.2. EXPORT TEXT DATA

Selecting this option will copy the scenario description and the orders for each side to a text file on your hard drive. You will be prompted to enter the path and file name. If you do not specify an extension, the file will be given a .SCT extension by default.



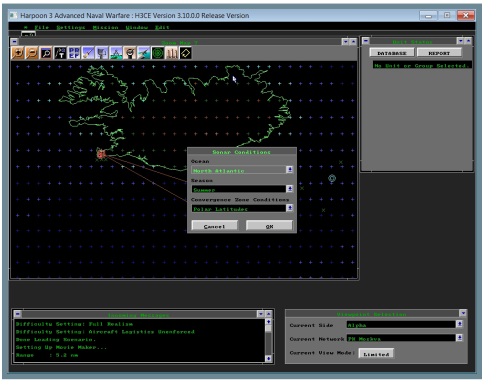
23.7.3. CHANGE START DATE AND TIME

Brings up the **Reset Start Date Dialog.** This dialog allows you to change the starting date and time. Note: Make sure you set a time prior to any Absolute scenario objective time.



23.7.4. EDIT SONAR CONDITIONS

This dialog allows the scenario author the ability to overlay sonar conditions from another ocean area on the current scenario. Customizations include selection of the Ocean area, the Season, and Latitude.

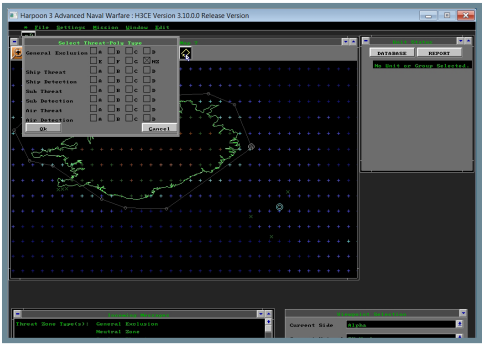


23.7.5. COPY THREAT POLY

To demonstrate the Copy Threat Poly feature, first, ensure that you have selected the Preferences tool bar icon and selected the **Neutral** and **General Exclusion** Nav Zones choices. Then **select**

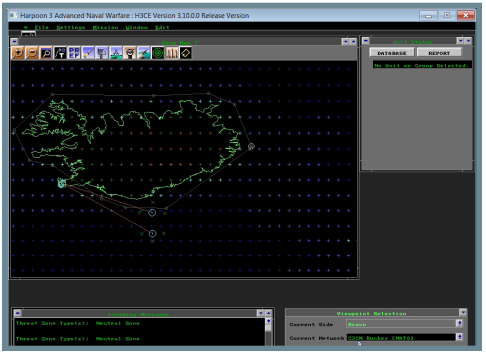
side Alpha, and click on the **Threat Poly toolbar button**. **Draw a poly** that follows Iceland's coastline, but cuts across the peninsula with the Bravo base so that it is not included in the poly. In this scenario, we want the combatants to respect the island's civilian populace.

The following is illustrative; you should be careful of how far south, try to keep it so that all three Alpha starting locations can draw a line to their target Reference Points off of Bravo's base.



Once the poly is drawn, **click on the NZ check box in the Select Threat-Poly Type dialog box**. This neutral zone designation will keep the players from editing or deleting the zone. The Other check boxes are used to record subtle restrictions for units when they are navigating through/into a Threat Poly.

Reminder: Each type of exclusion zone listed has several lettered boxes next to it allowing you to create multiple versions of the same type of zone, each with its own letter designator. For Example, you could create to Ship Exclusion Zones, one would be labeled "A", while the other could be designated "B".



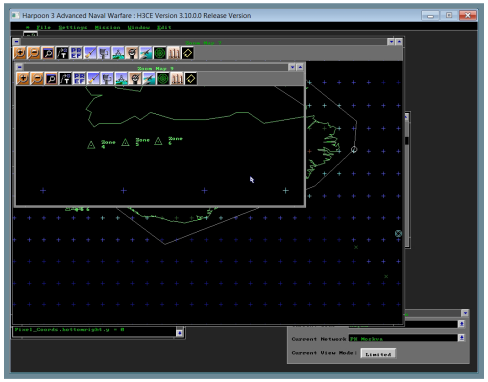
The Copy Threat Poly option allows you to copy the selected poly to all sides in a scenario. **Select side Bravo**. Notice there is no threat zone visible. **Select side Alpha**, **select the threat poly around Iceland**, and then **select Copy Threat Poly** from under the **Edit pull-down menu**. You should see messages informing you that it has been copied to the other sides. Now **switch to side Bravo**. Notice the threat poly is there. **Change to side Charlie**, notice it is there as well.

23.7.6. LOCK (UNLOCK) REFERENCE POINT

When a player starts a scenario, all 'unlocked' reference points are removed from the map. This allows the player to create their own missions with their own reference points. However, this poses a problem when Victory Conditions rely on a unit reaching an area marked by reference points for stations. These reference points must be 'locked' so that they remain on the map.

Make sure you have side **Alpha selected**, and then **open a zoom window** on the **Bravo base**. **Drag-select the three reference points to the south of the base**. As you will remember, these are the reference points used for Alpha's scenario objective. Now select Lock Reference Points from under the **Edit pull-down menu**. Then **select one of the points**. Make sure num-lock is off, and then hit **Ctrl + 7** on the numeric keypad.

You should see the **Rename Reference Point** dialog box. **Rename it "Zone**". Notice the name of the point has changed to Zone#. Rename the other two points as well.



If you wish to 'unlock' the reference points, you must select them, and then select Unlock **Reference Points** from the Edit pull-down menu.

NOTE: It is often useful to refer to reference points in the orders text as a way of 'nudging' the human player toward completing a scenario objective.

23.7.7. EDIT LINK16 SETTINGS

This feature is currently not implemented in Harpoon 3 ANW.

23.7.8. MISSION PROFILES (PROGRAMMABLE MISSIONS)

This is a newly (v3.10) added capability that radically enhances the ability of the scenario designer to engage the player. To further enhance the AI's performance, it is also possible to access parameters used to control the behavior of a mission and the mission's craft. With the parameters one can create mission profiles that stipulate specific attributes of a particular mission.

The menu items here are covered in more detail below, and are advanced even for this tutorial. Mission Profiles allow scenario designers to develop their own way of implementing different types of missions, and distribute those profiles as part of their scenarios. These menu options support that functionality but are not part of "basic" scenario editing/creation. There are outlined here for completeness.

Under the **Mission Profiles menu** item under the **Edit menu** there are two sub options, **Save** and **Join.**

Save profiles given to you from a scenario as an HPS text file. Normal scenario files (SCN files) include the HPS data when saved, this feature is simply used to deconstruct profiles shipped to you with a scenario.

Join is used when a designer has created a new profile based on an existing profile and the scenario is shipped out. If a player has their matching existing profile modified, the new profile isn't going to work. Thus, this feature "rolls up" the relationship (called inheritance for those geeks out there) to help minimize side effects.

A library of mission profiles made by others and instructions on how to use them is available on the Harpoon Wiki http://wiki.computerharpoon.com. Search for "Mission Profiles" .

23.7.9. UNIT MENU

The following describes the menu items not covered previously in the Tutorial.

23.7.10. UNIT MENU: SET PARENT UNIT

Used to set the parent unit (i.e. Base) of an aircraft placed on the map during editing of the scenario (vs having to launch them).

23.7.11. UNIT MENU: SCRUB LAUNCHES

When you set up formation air patrols in the Formation Editor, it sets the aircraft to 'launch' status. If you decide that you do not want this, you can select the Scrub Launches menu item. This will change the launch status to a 'ready', unassigned state for ALL the aircraft on the selected side.

23.7.12. UNIT MENU:REBUILD UNIT

These functions are only used when a unit has been changed in the database during the construction of a scenario. They are vestigial development tools. Use only under advice of the author of a database.

23.7.13. UNIT MENU:REBUILD UNIT MAGAZINE

Rebuilds a Unit's magazine to reflect the default weapons load from the current DB. Any Scenario specific Magazine edits will be lost unless the "Rebuild All Unit Magazines" command is applied.

23.7.14. SCENARIO REBUILD

23.7.15. SCENARIO REBUILD:EXPORT ALL UNIT WEAPON EDITS

This option allows for the export of all scenario specific edits to the default mount and magazine weapon records. This allows the scenario designer to load platforms with weapons available to his sides at the time of conflict. From the table, you can see when each model of the Harpoon came into service. So with a scenario date of say 1989, we should load USN ships with Harpoon IC or Harpoon IE SLAM. You modify the basic loadout in the Mk141 mounts normally say Harpoon IB to Harpoon IC, Export All Unit Weapon Edits, and when you use the import function later, you can maintain this edit to the basic loadout.

Country	Weapon (Model)	Year
USA	Harpoon (RGM-84A)	1977
USA	Harpoon (UGM-84B)	1977
USA	Harpoon IB (RGM-84C)	1982
USA	Harpoon IC (RGM-84D)	1985
USA	Harpoon IE SLAM (RGM-84E)	1984
USA	Harpoon IG (RGM-84G)	1997
USA	Harpoon II (RGM-84L)	2002
USA	Harpoon III (RGM-84M)	2010

23.7.16. SCENARIO REBUILD:REBUILD ALL UNITS

These functions are only used when a unit has been changed in the database during the construction of a scenario. They are vestigial development tools. Use only under advice of the author of a database.

23.7.17. SCENARIO REBUILD:REBUILD ALL UNIT MAGAZINES

Allows a player or scenario designer to make use of scenario specific weapon edits as mentioned above. It could also be used to create date and scenario specific loadouts for each scen based on the date.

23.7.18. SCENARIO REBUILD:IMPORT ALL UNIT WEAPON EDITS

Rebuilds all scenario designer modifications to the default database loadouts, mount weapon records and magazine weapon records.

23.7.19. BATCH REBUILD

Rebuilds all scenarios within a Battleset.

23.7.20. BATCH EXPORT WEAPON EDITS

Used to export all of the SEM files containing the scenario specific weapon record edits for all of the scenarios within a Battleset.

23.7.21. PLAYTESTING

Creating a successful scenario requires testing it for balance, challenge, and tempo. After you've played it a few times from each side, give it to some friends and get their feedback. The more tested the scenario is, the more likely it will be distributed. After all, the best scenario is an unknown scenario if you have a logic error in your orders that prevents players from getting to the good parts.

23.7.22. ORDERS

Now that the scenario has been created and we have discussed the miscellaneous features available for editing a scenario, it is time to enter orders for each side. Orders are entered in the right panel of the Side Maintenance dialog box. This part will be left up to you, however some conventions have been used in the past that you may find useful when writing your own scenario orders. These conventions include:

Header

```
ORDERS FOR CMDR BRAVO OPERATIONS

Intel/Sitrep:
========
Statements revealing the location of enemy units or
about the conflict layout in general

Mission:
=======
Statements based on scenario objectives

Execution:
========
Suggestions on strategy for accomplishing scenario
objectives; in addition, any absolute scenario
objectives should be mentioned in the orders:
“Occupy this area for 5 hours....”

Command and Signal:
===============
Name the flagship or location of headquarters and
state the required EMCON status (i.e. passive,
intermittent, etc.).
```

After you have entered the orders for each side (except Charlie), name your scenario and type in a text description. Again, this is up to you. If you lack inspiration, look at the name and scenario descriptions for other H3ANW scenarios.

23.7.23. SUMMARY

You have now completed all of the H3ANW Scenario Editor tutorials. Living up to the tradition of H3ANW, the Scenario Editor contains many nuances that were impossible to convey in this manual. Our hope is that, as you create you own scenarios and play those scenarios created by others; you will develop an appreciation for its level of detail, and its flexibility as a simulator of both naval warfare and the politics of modern warfare in general.

23.8. OTHER SCENARIO CONSIDERATIONS

23.8.1. MODELING COLLATERAL DAMAGE

The growing media coverage of wars has led the US military to adopt a new concept known as "surgical strikes." In opposition to carpet bombings of World War II, surgical strikes are made with "smart weapons," which are precisely hit their targets and often feature smaller warheads.

During the Gulf War, roughly 10% of the ordnance used was "Precision-Guided Munitions" (PGMs). Twelve years later, for Operation Iraqi Freedom, the percentage had risen to 90%. PGMs have allowed the reduction of the risk on population living in the vicinity of military facilities, key targets during air campaigns. When civilian lives or property are lost it is called "collateral damage". H3ANW can model collateral damage through the application of victory conditions.

Let's imagine a scenario taking place in Southern Iraq, in March 2003. There are the Rumalah oil fields (derricks, pumping and storage facilities). These belong to a side called "neutrals/civilians". If Iraqi Army units are stationed less than 1 mile from civilian facilities, there's a risk that the fields may be hit as well in case of strike against the military unit.

Using cluster weapons is highly hazardous to the neutral units in the target area, since cluster weapons are considered an area effect weapon. It is preferable to use "non-area effect" weapons when targeting enemy units in close proximity to neutral units.

If a singular warhead is a powerful one, its blast will spread to nearby structures/units, damaging them. You'll be notified of this when the evaluation comes at the end of the scenario: "You're facing court martial because unit X fired a weapon Y on the unit Z."

For a scenario designer, this feature can be very interesting to use. In the Iraqi scenario example. The overall goal of Coalition forces is to keep structures intact for faster rebuilding of Iraq. Implementing this as a Victory condition is particularly hard to achieve:

Protect unit: Type/subtype/class/ **unitname**

In this case, this would be: Protect unit: Facility/oil well (plus, the number of oil wells you want to protect) Say you have 4 oil wells "protected" out of 4. If one is destroyed, the player looses the scenario.

23.8.2. CENTRALIZED INTEGRATED AIR DEFENSE SYSTEMS AND HOW TO CONSTRUCT THEM IN H3ANW

23.8.2.1. WHAT IS A CENTRALIZED INTEGRATED AIR DEFENSE SYSTEM (IADS)?

A centralized IADS is a network of grouped sensors and air defense equipment that are interconnected via centralized Command and Control centers. Each group of sensors and equipment is responsible to defend their own airspace. They report their detections to a

centralized Command and Control (C2) center which in turn will report the data and issue commands back to the reporting group and all other air defense groups which need to know, in order to defend efficiently and effectively a nation's aerospace.

The Soviet Union/Russia and most of their client nations used this type of system throughout the Cold War. Given the vast territorial expanses needed to defend, and the sheer number of air defense equipment, these systems served these nations well. Historical successes include Vietnam, the Middle East in 1973, and the downing of numerous NATO aircraft (mainly spy flights) throughout the Cold War. The Gulf War stopped the impressive career streak of these systems, as Coalition forces were quick to destroy the Iraqi KARI system (a combination of exported Soviet and French systems), allowing complete air dominance over Iraq.

Many nations still use this system today, including North Korea, Syria, Iran, and other nations that depend on Cold War-era equipment. Most nations have attempted to improve their systems after the collapse of the Iraqi system in the first Gulf War but given the cost and logistics of such an endeavor many nations' air defense systems are still very centralized. There are also several NATO block or Western Allied members that use a centralized system structure, but they have had the resources to address the shortcomings of the system and capitalize on its strengths. These systems are thus still relevant to the world today, and their proper modeling in a high-fidelity simulation like Harpoon is also relevant.

23.8.2.2. WHAT ARE THE ADVANTAGES OF CENTRALIZED IADS?

The most obvious advantage is that the host nation will have multiple sources of information going to specific centers of command and control which can sort out the information and issue orders to each part of the system to effectively respond to any event. The second advantage is that a sensor in one part of the country does not have to be in range to know what is going on in another part of the country. IADS collects and distributes the information.

23.8.2.3. WHAT ARE THE DISADVANTAGES OF A CENTRALIZED IADS?

The key disadvantage is that if the centralized command and control centers or the communications channels are disrupted (or destroyed) the system becomes paralyzed. This is because each individual group is dependent on the command center above it. Data and commands no longer flow from one end of the air defense network to the other. This was proven during the first Gulf War, when Coalition forces specifically targeted command nodes of the Iraqi KARI air defense system. Billions of dollars of good air defense equipment (mostly French C3I nodes and export-Soviet radars & SAMs) became far less effective, and in some cases paralyzed, allowing Coalition air power superiority starting several hours into the war and throughout the rest of the combat.

23.8.2.4. CAN YOU REALLY IMPLEMENT THIS IN H3ANW?

H3ANW is well suited to model this type of network. Communications are modeled in the simulation, through the use of data-links and multiple side postures. **Data-links** are simply lines of communications between one unit and another. These are established by having communication-type equipment listed within their database entries. **Postures** are relationship settings that can be given to each side of play. They can be set to friendly, neutral and hostile. The posture settings are used here to dictate communication relationships between each side. Specifically, sides that are specified as friendly will share sensor data. Likewise, if the side is set as neutral or hostile it will not share the data. Given that you can create many sides with different postures, it is possible to create a relationship tree resembling a centralized IADS.

23.8.2.5. WHY SHOULD I IMPLEMENT THIS IN MY SCENARIOS?

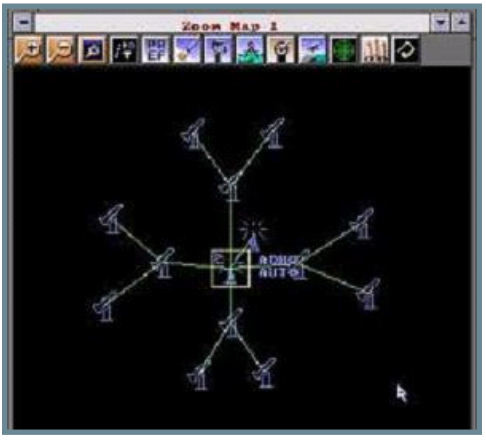
Simply put, to model reality. But the most important reason is that you are giving your scenario players a greater reward (or penalty) than merely a destroyed facility or platform. If the player intelligently picks his targets, he may be able to shut down an air defense system or sector entirely. This is a much more gratifying experience than simply destroying a single SAM. Likewise, reflecting the duality inherent in all warfare, the player's own air defense network can be shut down the same way, thus presenting a clear challenge (new headaches!) for the player. It is an implementation which really lends itself to a modern wargaming experience.

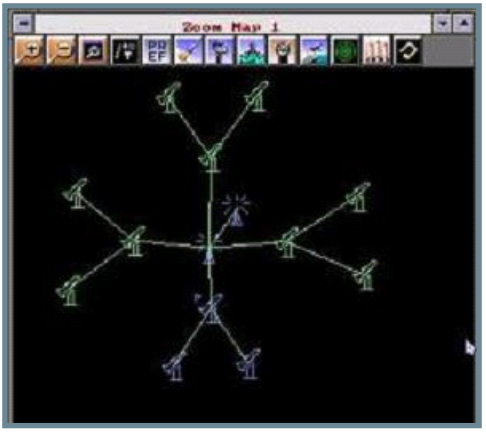
23.8.2.6. HOW DO I IMPLEMENT A CENTRALIZED IADS IN H3ANW?

Implementation is achieved by breaking down the air defense system into different sides, based on logical geographic or hierarchical groupings, and using postures to dictate the communication levels they have with each other. This fictional example is of a simple centralized IADS around the city of Bangalore, India.

The first step is to conceptualize a simple IADS for Banalore. One central command center is appropriate for the city with one long range EW radar attached. four subsectors are defined based on their geographic arrangement (North, South, East, West) around the city , each with their own side. Each subsector will include one SA-3 Regimental HQ unit and two SA-3 battalions.

Next, relationships are established between each of the sides to model their network. We specify the Southern Air Defense Command Air defense headquarters (ADHQ) as the central node, and thus should communicate will all subsectors and all subsectors should be able to communicate with it. To accomplish this, a “friendly” posture is set to all sectors and each sector side friendly to it. In order for the network to be centralized, none of the subsectors can be capable of talking to each other. This is accomplished by leaving their postures at “neutral”.

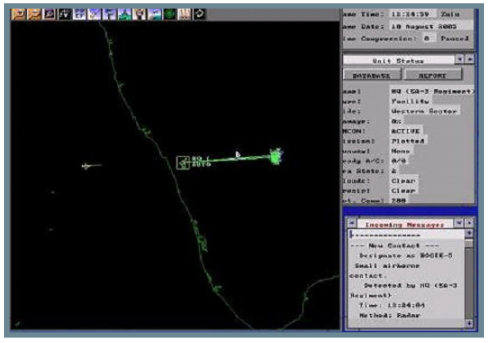




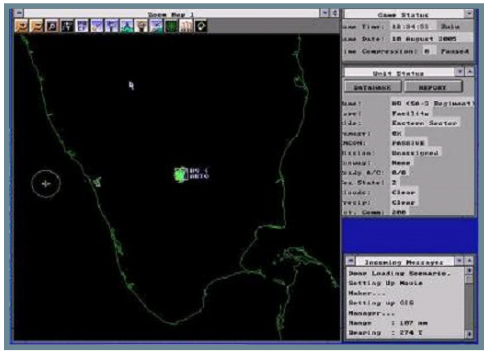
The screenshots show that each subsector components have datalinks that point to each respective sector HQ. Each respective sector HQ then points to the central command center node.

23.8.2.7. ENOUGH ALREADY, TEST IT AND SHOW ME IT WORKS!

In this example the western sector HQ has been moved west out of the range of the other sector's radars, so they do not see what the western sector HQ can see. A a "bogey" side is added, and assigned a single B-1B bomber, tasked to head east. To test this, switch to the Eastern sector side and wait to see a contact report is received from the western sector. At 12.24:04 the first contact report is received. The bogey was seen by the Western Sector's radars, and the report was passed through the central node and on to my current side, the Eastern Sector.



What happens when the Command Side is destroyed and all the links are broken between the sectors?



As you can see, the contact becomes lost as the Western sector no longer has a means to communicate with the Eastern Sector. All benefits of the interconnected IADS have been lost

and any bogie will now have a greater chance of successful penetration, as the component parts of the IADS are not communicating.

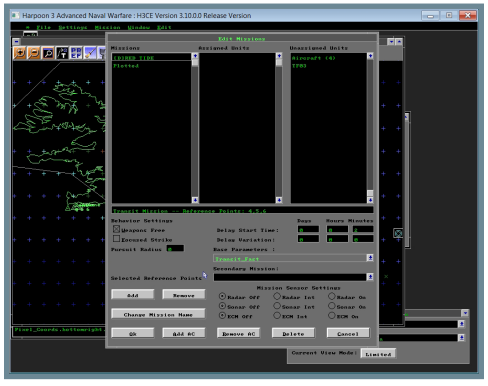
This was a very simple example, but you can build a huge and complex network using these principles. Just remember to break the system down into different sides and make sure that you use postures correctly. The command side posture is friendly to every sector and every sector friendly to it. Each sector is friendly to the command side but neutral to each other. The Central Node is friendly to everybody, and every sector is friendly to it. Each Sector is friendly to the Central Node but Neutral to all others.

23.8.2.8. ONE CAVEAT

In the instance that you have friendly aircraft sharing airspace with a hostile that is being engaged, it is technically possible that the friendly may interpret the friendly attack as being itself attacked. The result would be the assumption of a hostile posture. Careful planning and management can prevent this.

23.8.2.9. USING A PROGRAMMED MISSION PROFILE IN THE GAME

When editing a mission you may select a **Mission Profile** for any mission except for the plotted mission.



Mission profiles are saved in the Doctrine directory and any properly formatted *.hps file therein will be loaded when the game engine is loaded. A dictionary of the mission parameters is available from the Harpoon Wiki. (see http://www.computerharpoon.com) for links to the Wiki.

23.8.2.10. INHERITANCE

The mission profiles can utilize a programming concept called *inheritance*. This allows a new mission profile to start with an existing base profile and "override" it as necessary.

The following demonstrates a custom patrol mission profile using the standard patrol mission profile as its base. Without further parameter definitions this profile would be effectively identical with the standard patrol mission profile. The **Join menu** item was developed to assist with this issue.

```
[PropertySet]
Name=CustomPatrol
Inherit=MissionDefault_Patrol
PropertyName1=MissionParameterSet
PropertyType1=Null
PropertyValue1=
```

The key difference here is to both inherit the standard profile and use the name of the standard profile. The standard profile will maintain its existing settings with the changed settings in effect. Care should be taken when overriding the standard profiles as this will affect all missions by default. Further, overriding settings in a profile that is inherited by another will affect both.

This example happens to override the auto defense setting to disable it for the Weapons Free ROE.

```
[PropertySet]
Name=MissionDefault_Patrol
Inherit=MissionDefault_Patrol
PropertyName1=MissionParameterSet
PropertyType1=Null
PropertyValue1=
Propertyname2=MissionAutoDefenseWeaponsFree
PropertyType2=Integer
PropertyValue2=0
```

23.8.2.11. FILE AND FORMAT DETAILS

The mission profiles are defined in a .hps file in the Doctrine directory located as defined in the Harpoon3.ini file.

```
[PropertySet] - This is the first tag of the .hps file
and part of the mission profile definition. This line
will be the same for all mission profile files.
Name= - This is the second tag of any mission profile
and should give the profile a unique name to be
displayed to the user.
Inherit= - This tag is optional and can set a base
for the mission profile. It specifies the name of the
base profile from which it inherits.
Each parameter consists of three parts and a unique
number ID. The "#" represents the ID and should be
the same value for each part of the following tags:
PropertyName# - This sets the unique name of the
parameter.
PropertyType# - This sets the type of the parameter.
It will be Null, Integer, Real (Decimal), or Text.
PropertyValue# - This sets the value of the
parameter. It should be compatible with the type
specified or will lead to unpredictable game
behavior.
```

23.8.2.12. SAVING, LOADING, AND JOINING

Using the menu item **Edit->Mission Profiles->Save** to save each mission profile to the Doctrine directory from the presently loaded scenario.

The **Join** menu item is a way to create unique mission profiles from existing custom profiles. The scenario editor user should use this feature when they want to prevent profiles that the end-user creates from overriding the editor's own custom profiles. The command will create a new profile with the name based on the side, profile name, and mission name of the first mission encountered that uses that profile.

23.8.2.13. EXAMPLES

The Minimal Profile

```
[PropertySet]
Name=CustomPatrol
Inherit=MissionDefault_Patrol
PropertyName1=MissionParameterSet
PropertyType1=Null
PropertyValue1=
```

Boarding and Takeover Patrol

```
[PropertySet]
Name=PatrolBoardingTakeover
Inherit=MissionDefault_Patrol
PropertyName1=MissionParameterSet
PropertyType1=Null
PropertyValue1=
PropertyName2=Mission_Board
PropertyType2=Integer
PropertyValue2=1
PropertyName3=Mission_BoardTakeover
PropertyType3=Integer
PropertyValue3=1
```

A note about boarding and takeover: Strike and patrol missions may be set to board and takeover other craft. Strike missions used this way allow for specific targets. Patrol missions allow for randomly selected targets. To set a mission to do so you must select a mission profile set to board and/or takeover.

23.9. SCENARIO DESIGN GUIDED CHECKLIST

This is a list of suggested procedures and content for creating a scenario.

23.9.1. DESIGN

- Who are the groups in conflict?
- Why are they in conflict?
- Where does the conflict take place?
- How will the conflict be resolved?

23.9.2. MAP SIZE/GAME TEMPO

- **Quick:** 5x5 degrees, littoral surface conflict
- **Average:** 10x10 degrees, limited air assets, surface groups

» **Slow:** 20x20 degrees, long range air assets, carrier groups

» **Very Slow:** 50x50 degrees, multiple air bases, carrier groups with long transit times.

Additionally, game speed depends on the number of units and the size/type of map]. A big scenario with more land and less units will run as fast as a smaller map with less land and more units.

23.9.3. ORDER OF BATTLE

Listing of assets to include in scenario

23.9.4. MAP GENERATION

» Consider game tempo

» Cannot exceed 179 degrees in width (Latitude) and 180 degrees in length Longitude.

» West Longitude and South Latitudes are negative. East Longitudes and North Latitudes use positive values.

» It should be noted that the weather is created with the map. If the user disagrees with the forecast then the map will have to be recreated. Weather is probably the very first thing to be decided upon in a scenario. Additional information about weather in Harpoon and how it can be implemented in a scenario is available on the Harpoon Wiki.

» Sonar conditions can be set independently.

23.9.5. ADDING SIDES

» Postures

» Placeholders for orders

23.9.6. ADDING UNITS

» Bases are auto-detected

» Use the **Edit Aircraft menu** to place grounded AC

23.9.7. MISSIONS

» Always time delay transit and area missions by 1 min.

23.9.8. VICTORY CONDITIONS

» Relative times start when the designated unit reaches the specified waypoints.

» Absolute starts at time in start field

» Delta time is a contiguous duration

» Edit Scenario Duration? (Note that it is rare for scenarios to proceed beyond 3 days in length.)

» Reset Elapsed Time

23.9.9. MISCELLANEOUS

» Copy Threat Poly?

» Lock/Unlock Reference Points?

» Include Orders and Scenario Description.

24. DATABASE EDITING

It is the function of the Navy to carry the war to the enemy, so that it will not be fought on U.S. soil. - Fleet Admiral Chester W. Nimitz

AGSI and Matrix do not support database editing, these tools are provided "as is" without any warranty or support for the general public. AGSI will, in select circumstances, enter into agreements with database authors to provide them with assistance in developing and maintaining databases that ship with the game. In these select circumstances, AGSI will provide data from Mr. Bond and Capt Carlson as well as database tools to support the creation of content.

The key to H3ANW flexibility is its database (sometimes referred to as 'DB'). The DB, which is modular in design, provides the user with all the information necessary to make playing the game as real as it is. H3ANW would not have existed if it wasn't for the flexibility of the DB and its DB editors.

The current H3ANW Reimer Editor (H3RE) has its origins in two previous editors.

The first was the original H2 DB editor created by Three Sixty Pacific (TSPI), the original developers and publishers of Harpoon II (H2), the precursor to H3ANW. The other editor was called PfEdit created by Tom Wenck. It was a simple 16-bit DOS application that saw widespread use.

Both editors eventually fell victim to advances in computer technology. The original TSPI editor was cumbersome to use and required a screen resolution beyond the capabilities of most computers of the era. PfEdit, was limited by the 16-bit nature of the program and was limited to the 640kB conventional memory of pre-Win95 operating systems.

In the mid-1990s, Jon Remier, another H2 player set out to create a new editor using MS Access and Visual Basic. He was able to overcome all the limitations of the other editors and create a program that ran in MS Office. This editor is the most widely used means to edit the various H3ANW (and Harpoon Commander's Edition) databases. This section of the manual will focus on this editor, to explain the concepts needed to create/edit databases. AGSI obtained the rights to this Editor in 2005 and named it the "Reimer Editor" in honor of Jon's work.

There is also an editor based on the H3ANW Game Engine (H3CD).

H3RE also contains a DLL file that will generate the DB signature automatically when it exports the entire DB as implemented in v3.10 of H3ANW.

There have been almost a dozen user developed databases, The only databases supported for v3.10 bug reporting are ANWDB and HUD3 (both are updates to the "Original DB" or ODB).

» HUD3 by Darren Buckley (and crew)

» ANW (or ANWDB) by Dale Hillier

The reason for this is the game engine (i.e. simulation) effectively consumes the database data as "run time" source code. If a value is used differently than the simulation was programmed to accommodate, the simulation will not run as AGSI intended. This power has been used to extend the simulation and to overcome programming defects in the past. The problem is that when the game engine evolved, the database authors didn't want to! Keeping backwards compatibility meant maintaining more and more source code, more testing, and more support. Rather, AGSI now provides older editions of the game engine to players so they can enjoy their work for as long as MS Windows allows, and give them time to upgrade to the newest simulation features.

24.1. ACCESSING THE EDITORS

The Launcher will allow you to launch the Game Based Editor (H3CD) after you have installed H3ANW. However, **there is one caveat – it only runs at resolutions of 1024x768.**

H3RE requires the free Microsoft Access runtimes or the full Microsoft Access license.

These instructions are H3RE specific as most players will use this tool over the H3CD for most editing operations, except for the encryption feature which must be executed by the H3CD product.

24.2. DIFFERENCES BETWEEN H3CD AND H3RE

As of v3.10 H3CD can encrypt a Database, H3RE can't.

24.3. CREATING A NEW DATABASE

The player can simply copy the named database folder under the installation tree in the databases folder, or they can follow two methods for use with the H3RE.

The first method is to copy the MDB file and give it another name. This doesn't involve the Access editor per se because the editor only modifies the Database (DB) files when it exports (see below). However you will have to make another copy of the MDB file to use for the new database. To create a new DB:

» Create a folder for the DB in your H3ANW installation

» Copy a DB file set into this folder. It doesn't really matter what DB, it's all about personal preference; i.e. copy the HUD3 files to a 'new DB' folder, or whatever name you wish.

» Copy the .MDB file and rename it to whatever your new DB is going to be called. (i.e. copy the ANWDB.MDB file and call it **Personal.MDB** or whatever).

» Run the new MDB file and the editor will appear. It will contain the data from the copied MDB file.

The second method is to use the 'Erase Database' function in the main menu of H3RE. This will clear the DB of all platforms leaving you with a blank DB. This does not involve renaming the .MDB file but, for obvious reasons, scenarios that you made with the DB you just erased will be unplayable. Most DB authors like to start with some sort of structure in place already (as was the case with the Colonial Wars DB and the 1960's-1970's DB, both of which were legitimate copies of Mr. Ragnar Emsoy's DB2000..

IN ALL CASES THE COPYRIGHT TO THE DATABASES RESIDES WITH THE DATABASE AUTHORS. The encryption feature was added to provide some security for database authors.

You now have the tools necessary to start your new DB. If you wish, you can erase the database totally and start from scratch, or you can selectively delete various platforms so as to keep the integral parts of the DB (such as sensors and weapons) intact and thus save you a lot of time.

Make sure that your MDB import/export default file path is set to the correct folder otherwise you could import over another DB and lose that information; i.e. make sure that the MDB file exports to C:\Harpoon 3\databases\my DB instead of C:\Harpoon 3\databases\HUDIII.

24.4. IMPORTING A DATABASE – H3RE

The H3RE editor works by importing an ANWDB. The editor will ask you to point it to the correct directory. It's important to note that when importing the DB the editor only READS the files and doesn't make any changes to them.

To import a DB, click on the **Import New Database** button in the menu form and it will then bring up a small window with 2 options and a start button. The first option is to use the default directory. The second option is to import text. Both of these will be explained later.

When you hit start, the editor will bring up a browse listing and you will then have to select the folder where the DB resides in. The files are found in the database folders inside your H3ANW/databases directory and carry a .dat extension.

It is important to note that when you import a new DB, you'll be erasing any other information that resided in the editor (not the source file) at that time. After you select the folder the process is automatic from there. The editor will then read all DB files and process them.

After the import is complete the editor will advise you to compact the DB to make the whole process run faster. The editor will provide you with instructions on how to do this.

You will then be ready to read and modify the DB as you wish.

It's also important to note that when you are looking or modifying the DB, you are not changing the DB files itself, only what's been imported into the editor. To make changes to the DB files themselves you'll have to export the files.

24.5. EXPORTING A DATABASE – H3RE

When you export the DB it will create a series of .dat files that will overwrite anything in the target folder.

24.6. ENCRYPTING A DATABASE – H3CD

Database Encryption uses the v3.10 H3CD and a password to encrypt a DB into a single file to be used by the game engine.

Other users will be able to create, edit, and play scenarios with the encrypted database; however editing the database will be impossible without the password.

24.6.1. CONVERTING A DATABASE FROM THE ORIGINAL .DAT FORMAT FOR ENCRYPTION

1. Backup the database that will be encrypted.
2. Launch H3CD; this will automatically load your database from its folder
3. Save the database using the: **Database->Save Annexes** command. This will convert the existing database format to v3.10.
4. Exit the editor
5. Two new files will be created in the folder locating your database files, 'signature.bin' and 'annex.hsd'. These two files are the complete encrypted database (modified text and picture files are not included in the encryption).

24.6.2. SETTING THE PASSWORD FOR THE ENCRYPTED DATABASE

1. Run the v3.10 H3CD with a v3.10 compliant database
2. Select the **Database->Secure Database** drop down menu. That opens a submenu; select '**Set Database Password'**.
3. A small pop-up window will appear requesting you type in the password. Once you've entered the password, hit **Enter**.
4. **Save** the database, this is required to finish the encryption.

The database can now only be unlocked using the new password. This gets us to our warning:

Remember your password. There is not, nor will there ever be, a password recovery system for encrypted databases. Think wisely about your password. Use something original. If you forget/lose your password, you will never be able to reopen the database. AGSI cannot and will not be responsible for lost passwords.

24.6.3. UNLOCKING A PROTECTED DATABASE FOR EDITING

1. Run the v3.10 Database Editor with a v3.10 compliant database

2. Select the '**Database'** drop down menu and select '**Secure Database'** from it. That opens a submenu; select '**Un-Lock Secure Database'** from it.
3. Enter the Password
4. The program will inform you of the success or failure of the procedure.
5. After a successful un-lock you will be able to change the password using the same procedure to set it. You can also clear the password by not entering anything and hitting '**ok'**.

A locked/unlocked indicator has been added to the user interface to indicate the status of the encrypted DB. At a later date, a password file will be added so that a user need not enter it every time. The file will NOT be in the same folder as the database to prevent the accidental distribution of the password.

The editor creates the HSD file as the encrypted database. This HSD file is what is to be distributed to the public. The DAT files are to be kept by the DB author for editing by the H3RE editor. The H3RE will still be functional and will use the .DAT files. Remember that it's the single HSD file, as created by H3CD, that is to be distributed and NOT the .DAT files.

AGSI cannot and will not be responsible for the accidental distribution of the DAT files.

24.7. OTHER DATABASE BUILDER FEATURES

Opening any annex will cause the editor to create a new entry in the tool bar at the top of the window. This entry is called 'Harpoon' and it contains a variety of useful tools that can assist in speeding up the DB creation process.

Copy Item – This will have the editor make an exact duplicate of whatever you are viewing (platform or component) at the time. The new item will be an exact duplicate of the previous item with the exception that will have a different version number. This is useful when making platforms of the same design but for different countries.

Import – This command will bring up a submenu that gives the user the choice to import a totally new DB, annex (one of the .dat files) or an HCF file. You will be prompted for a file name and/or location.

Export – This command is the opposite of the Import command. It will bring up a submenu that will allow the user to export a DB or to a directory, or save a platform or component as an HCF file. You will be prompted for a file name and/or location.

Add to Export List – When getting ready to create an HCF file, using this command will add whatever you are looking at, to a list of items to be converted into HCF files.

Clear Export List – This will clear the HCF export list.

Recalc Unused – This useful command will have the editor scan the annex that you are currently viewing and find any items in the annex that are not currently used by any other annex. Items that are unused will lose the check mark located at the far right side of the screen.

Show Unused – A simple filter command that will show a list of all un-used items in the annex the user is viewing.

Delete Unused – This command will, after user confirmation, delete all unused items in the currently viewed annex.

Validation Report – This command will generate a text file of all errors and warnings in the annex that you are currently viewing. This is one of the more important features of the editor, as it will show you a list of platform names that are too long. Platforms with this error will cause the game to crash. It's important to use this on every annex on a regular basis.

Show Platform – A simple secondary menu that will allow the user to change from one annex to another or open an unopened platform annex.

Show Component – Another simple secondary menu that will allow the user to change from one annex to another or open an unopened component annex.

25. PLATFORM FIELDS

"It follows then as certain as night succeeds day, that without a decisive naval force we can do nothing definitive, and that with it everything honorable and glorious." - George Washington

This section will clarify some of the definitions, fields, and annexes associated with various unit types and applies to both editors.

Warning: AGSI does not support database editing or creation. This is an extremely advanced topic that directly impacts how the simulation runs.

25.1. EDITING PLATFORMS

Platforms have a variety of entries that you can modify. The dimensions of the platform are only for the platform database entry in game. In addition, platform dimensions can affect the platforms cross section, but only if the player hits the **automatic cross-section button**. It is only when you get into the platform buttons **(Flags, X-Section, fuel, etc)** that you'll be altering a platform's performance in the simulation.

All measurements use the metric system. Meters, Kilograms, and Metric Tonnes (called Tonnes or Tons). All speeds use knots (Nautical Miles per Hour) and ranges are in Nautical Miles.

25.1.1. AIRCRAFT

When editing aircraft entries in the databases, there are a variety of values (aka attributes) that can be changed. This section will explain the different settings and their effects.

25.1.1.1. MISC

Length, Wingspan, and Crew are self-explanatory.

Weight should show the empty weight of the aircraft.

Climb Rate is usually found in many reference books. If it's unavailable simply take the cruise speed of the aircraft and multiply it by the sine of 45. Rate of climb is currently limited to 255 m/s.

Climb Rate = cruise speed (meters per second) x sin 45

25.1.1.2. ATA

This is probably the most important aspect of the most aircraft in the DB. Simply put, the higher the ATA value, the more maneuverable that plane will be. It is certainly the most important aspect for fighters and attack aircraft. ATA can also be described as the ability of a missile or aircraft to avoid another missile or aircraft. For aircraft this is value is used so the H3ANW game engine can calculate ACM between aircraft and an aircraft's ability to dodge missile or gunfire.

The ATA is a comparison between an attacking platform and a defending platform. It is a generational comparing between other aircraft and weapons of its generation. It is purely subjective and assumes that both pilots have the same skill.

As an example, an SA-9 might be able to hit a B-52 75% of the time, and an F-16 only 30% of the time. Given equal ATA values (e.g. both missile and aircraft are ATA 5), the Probably of Kill (PK) of a weapon will be a base value of 30% before any other modifiers like jammers and decoys come into play. Any changes of 0.5 ATA equate to 5% PK. Thus if the SA-7 has a 35% hit rate against an aircraft like the F-15 with an ATA of 4.5, then the SA-7 will have an ATA rating of 5.0.

The Harpoon 4.x Master Rules' ATA combat table has the following differential values 0.0 = 30% 0.5 = 35% 1.0 = 40% ... 5.5 = 85%.

For numbers above 0.0 you get a conversion formula: (ATA/0.5) x 5% + 30% = PK%

However, the PK numbers for the missile might be geared towards a specific type of target, which might mean something too: a typical fighter has a D-ATA of 4.0/2.0 while a large A/C like the Sentry has 0.5/0.5

25.1.1.3. AIRCRAFT FLAGS

Probe Refueling – Indicates the aircraft has a probe for mid-air refueling; most aircraft in the world are probe refueled.

Centerline Drogue – Indicates that the aircraft has a hose with a drogue at the end of it to refuel aircraft fitted with refueling probes. The hose is reeled out of the back end of the aircraft. Some Royal Air Force (RAF) tankers have this fitting.

Wing Drogue – Indicates that it has hose and drogue gear at either wing tip. The KC-130 has this type of refueling gear.

Boom Refueling – The aircraft in question requires rigid boom refueling gear to accomplish its mid-air refueling. All United States Air Force (USAF) aircraft use this method of refueling.

Centerline Boom – Indicates that the aircraft has boom type refueling gear; the KC-10 and KC-135 have these attachments. It's important to note that the boom gear can be fitted with a short hose and drogue to refuel aircraft fitted with a probe.

Terrain Avoidance – Aircraft that have the ability to fly low enough to avoid terrain but not match flight profiles with it. Aircraft with the flag cannot fly lower than 100m in game.

Terrain Following – The aircraft has a sophisticated computer that allows it to fly so low that it can match its flight path with the terrain. Aircraft with this flag cannot fly any lower than 40m.

Bombsight – There are three types of bombsights: Basic, Computing, and Advanced.

Helo Pad – This is a size factor. There are three types: Small, Medium, and Large. A small helo can land on all of them, while a large helo can only land on the Large Helo Pad and nothing else.

Runway – Like the helo pad flag, this too is based on size. VTOL runway is pretty much the same has a helo pad. STOL is for aircraft with short take off and landing capabilities. Aside from that the other size parameters are the same.

Carrier Capable – Indicates the aircraft is capable of taking off and landing on an aircraft carrier. (Note that in many databases this flag is set for all aircraft to allow fictional airbases at sea)

Aircraft Size – Based on aircraft size. Refer to section 25.6 Air Facilities for further details.

Blip Enhance – This is mainly a helo flag. It allows the aircraft to broadcast a signal that makes it appear to be a medium sized (or larger) surface vessel. Used for decoying anti-ship missiles.

Helicopter – This flag tells the game engine that the aircraft is a helicopter and is capable of hovering.

25.1.1.4. SHOW RANGES BUTTON

This will display a small window that shows the ranges for all the loadouts that the aircraft carries.

25.1.2. FACILITIES

Facilities are ground based platforms. Facilities have several fields that are important.

Mast Height - This is an indicator of how far the sensors on the facility can see. The higher you are, the farther your visual horizon is. The same principle applies to radar as well.

Missile Defense Rating (MDR) – This is a calculation of how well a platform can defend against incoming missiles. To determine the MDR for platforms (it also applies to ships as well) proceed with the following: during combat, this number is used by the Artificial Opponent (AO) to determine the number of missiles enemy units will fire on this unit based on an assumed level of gathered intelligence. This number should be based on the number of fire-control channels available for anti-missile fire.

25.1.2.1. AREA DEFENSE SYSTEMS

(10 nm or greater anti-air range)

MDR = # shots vs. a mach 1 closing target x number fire-control channels x time pK of weapon.

25.1.2.2. POINT DEFENSE SYSTEMS

(<10nm anti-air range)

MDR = number of firing arcs / 8

25.1.2.3. DISTRACTION SYSTEMS

+1 if Defensive ECM system present

+1 if Chaff/Flare launcher present

25.1.3. ARMOR RATINGS

These apply to ships as well. There are four levels. Light, Medium, Heavy and Special. According to the H4.x Master Rules, the definitions of each level are as follows:

Light – 90 to 120mm RHA (Rolled Homogeneous Armor)

Medium – 121 to 140mm RHA

Heavy – 141 to 190mm RHA

Special – 191 to 406mm RHA

These values are highly subjective, if for the only reason that there's a difference in materials used (Concrete vs. Steel, for example). You'll have to use your best judgment in that regard.

25.1.4. A NOTE ABOUT DP FOR FACILITIES

Facility DP is based on the number of weapons it would take to destroy that facility. An M-1 tank would require a Maverick, TOW, or Hellfire missile to destroy it. Therefore, if a Maverick has a DP value 15, then an M-1 tank platoon (four tanks) would have a DP value of 60 points.

The obvious problem here is that other weapons have greater than 60DP but can still attack a fixed facility such as a Tank Platoon. The use of facilities in this regard would help but increase the size of the scenario. The bottom line is that H3ANW isn't meant to model tactical combat on that level at this time.

Of course, different warhead types will inflict different damage levels based on their type. A case in point is a 500 lb GP bomb will probably not penetrate the armor of a WWII ship, but a 500lb AP bomb may penetrate and cause internal damage.

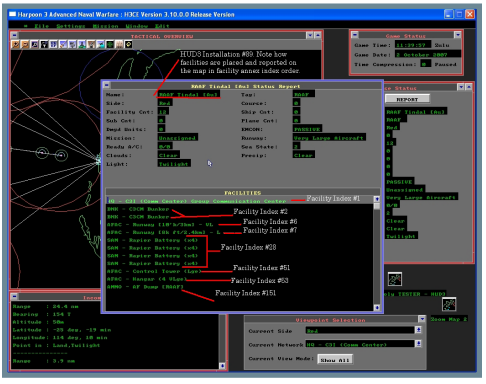
25.1.5. INSTALLATIONS

An installation is a group of facilities lumped together. These are effectively Task Forces (aka Groups) for facilities such as SAM batteries, a Coastal Artillery Site, and Radar Installations.

This is where the Offset comes into play. The offset allows the game engine to separate units and spread them around. In extreme cases, you can accurately model what an installation looks like in real life.

25.2. BASE/FACILITY/INSTALLATIONS ISSUES

Weapons that require command guidance or employ semi-active homing methods will require that the guidance sensor (or comm unit) be placed **on** the launching facility. This causes problems for units like the SA-2, which has 6 launchers with a radar and command van in the middle. However, active homing weapons such as CLAWS (ground launched AMRAAM) or coastal systems like Exocet will not be affected.

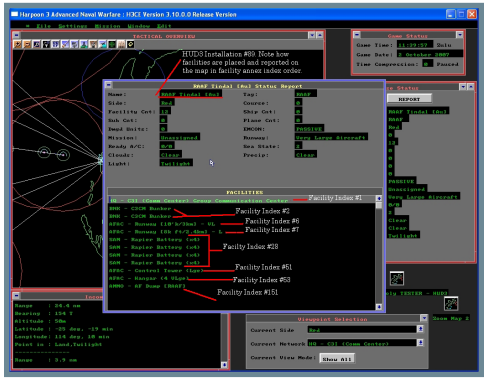


The second involves air bases. The game engine does not reliably report the total number of aircraft in a scenario which includes air bases included in Installations. This happens because the game engine doesn't count the facilities in the installation in the same way as they would if the scenario builder created each facility individually.

There is a work-around. It is believed that the anomaly occurs when the installation is actually placed.

Because an installation is grouped, and because the game engine DOES 'plant' the facilities in a certain order, then the work-around for this is to delete the group. This will remove the 'base' so to speak but not the facilities that are in it. Then you can regroup the base normally.

HUD3 has the HQ – C3I (Comm Center) facility as facility index #1 This makes the comm center the the group command center, and should overcome all reported missing units. A good practice is to add most AAA and SAM as standalone units, and then hide them by using the Home key (reference the Scenario Editing Tutorial and automatic Base detection).



25.3. SHIPS

These designators impact various aspects of the simulation engine, including how damage is applied.

Type – This is an up to 5 letter acronym used to indicate the type of vessel. It is required to create a new ship but doesn't affect the game in any way except for platform type during ship identification.

Category – This is meant to signify the type of ship for the game engine. There are five classes: Surface Combatant, Merchant, Carrier, Amphibious, and Auxiliary. It is required by the program engine for determining the types of applicable icon the game displays for a given vessel.

Surface Combatant – This applies to Cruisers, Destroyers, Frigates, and anything with guns/missiles manned by military personnel that floats.

Merchant – These are civilian owned ships. Usually they are very easy to destroy.

Carrier – Any ship designed to carry and launch fixed wing aircraft.

Amphibious – Any vessel that has been designed to support the landing of men and material on hostile shores.

Auxiliary – Any vessel designed to support naval vessels (UnRep vessels, ammo ships, oilers, etc).

25.3.1. SHIP ANNEX FIELDS

Length – The length of the ship (in meters).

Draft - The depth of a vessel's keel below the surface (in meters) – Reimer Editor Only

Beam – The breadth/width of the vessel at its maximum point (in meters) – Reimer Editor Only.

These values are used to automatically calculate the platforms cross section in H3RE. In H3DC you will have to manually input the cross section fields from data developed using the 'H3 Platform Assistant.'

Displacement – Defined as the weight of the water displaced by the ships hull form, this is expressed in metric tons. Whenever possible this should be the lightship displacement which is the displacement of the vessel when totally empty (the technical description of lightship is the displacement of the vessel when empty with the exception of lube oil for the main engine).

Crew – The number of men required to operate the vessel.

Troop Capacity – The number of armed troops the ship can carry (not crew). This option is not currently supported.

Cargo Capacity – The amount of cargo, in metric tons, that the vessel can carry. This option is not currently supported.

Damage Points – This is a numerical value of the ship's survivability in a combat situation. It's directly related to light displacement. The formulas for calculating vessel DP is as follows:

Displacement (in tons)	**Formula**
0-500	Tons/20
501-5000	(Tons /30) + 9
5001-12,000	(Tons /50) + 76
12,000 and over	(Tons /60) + 116

Once you have determined the damage points, multiply the value by any of the following multipliers that apply:

- » Fleet Auxiliary (supply ships, container ships, tankers, ammunition vessels, etc.), not including converted merchant vessels – 0.75
- » Surface-Effect Ship (SES), Hovercraft, or Merchant (including all ships specified as being built to civilian standards) – 0.5
- » Supertankers – 0.25
- » Soviet Construction – 0.9
- » Composites; Aluminum, GRP, Wood – 0.75

The following attributes are unrelated to Damage Points:

Max Sea State – This is the maximum seas that the vessel can sustain combat operations. Not Supported.

Fuel Transfer Rate – The rate, in tons per minute, that fuel is transferred. It is used for UnRep.

Missile Defense – See Missile Defense in the Facilities Section

Armor Ratings – See Armor Ratings in the Facilities Section

25.3.2. SHIP DATA FLAGS

Refuel Astern In/Out – This flag signifies that the vessel is capable of accepting or pumping fuel from another ship. It is an older method of refueling that is common among older tankers and civilian auxiliaries.

Refuel To/From Port/Starboard – Indicates that the vessel can accept from a ship or pump fuel to another ship. Typical UnRep ships have four fueling stations, two on each side.

Replenish To/From Port/Starboard – This flag is similar to the Refuel flag but applies to weapons instead of fuel.

Passive/Single Stabilizers – This indicates that the ship has either a passive (flume tank) or single pair of stabilizers.

Dual or Triple Stabilizers – Ships with this flag have multiple or active sets of stabilizers.

Nuclear Shock Resistant – This indicates that the vessel in question is built to withstand massive shock waves such as those created by nuclear weapons. This flag generally only applies to older battleships and cruisers. Is used in calculating damage effects.

Civilian Construction Standards – Used to indicate that the ship is not compartmented or strengthened with multiple watertight internal bulkheads in a manner similar to warships. Is used in calculating damage effects.

Aviation Vessel – Aside from the carrier flag mentioned above, this flag is used to indicate vessels that can support more than the usual number of helicopters (1 or 2) carried on a warship. Examples of this are the Japanese Navy DDH or the Italian Vittorio Veneto. Is used in calculating damage effects.

Aluminum Superstructure – Aluminum superstructures were a feature of 1970's ship design. Indeed, they still are for many merchant ships. However in the Falklands War of 1982, it was discovered that Aluminum also melted at a much lower temperature than steel. This led to a basic shift in warship design back to steel superstructures. The flag mainly applies to ships built in the 1970's and 80's. Is used in calculating damage effects.

25.4. SUBMARINES

Type – This is an up to four-letter acronym that is used to denote the type of platform. It is required to create a new sub in the editor but doesn't affect the game in any way.

Length – The length of the submarine.

Displacement – This is similar to the ship entry (except that in some circles a 0.5 DP modifier is applied due to the relatively small reserve buoyancy of submarines).

Crew – The number of crew on the submarine.

Maximum Operational Depth – Crush depth may be deeper, but H3 ANW only uses this value. It must be a negative value.

Damage Points – See above for ships.

25.4.1. SUBMARINE DATA FLAGS

Anechoic Coating – This flag is for subs that have either a layer of applied composite or tiles that are used to absorb sounds and prevent detection by sonar. Used by most modern subs.

Nonmagnetic Hull – Subs using this flag have been built using non-magnetic materials. Russians subs such as the Alfa and Sierra classes (with Titanium hulls) are examples of this.

Double Hull – Mainly a feature of Russian subs, this flag indicates the presence of a second pressure hull. This makes subs more resistant to damage.

Shock Resistant – This is similar to the ship flag of the same name.

No Launch Transient – Subs with this flag do not radiate the sounds associated with the launch of a weapon. Only the most modern submarines have this flag.

Shrouded Propulsor – This is a design feature that lowers the passive sonar signature of the sub. It indicates that the sub has a directional collar around it's propeller (on merchant ships this is called a Kort Nozzle). This feature is not currently supported in the game.

Advanced Propulsor – Used to indicate the presence of a pump jet propulsor. It is unknown if this feature is not currently supported.

Snorkel – A universal feature in almost all subs, this allows the sub to run on diesel engines when they are at periscope depth to re-charge their batteries.

25.5. COUNTRIES

This is an organizational listing of platforms listed by country. The editor will guide you along the process of adding the DB ID numbers.

25.6. AIR FACILITIES

This component of the DB contains all the information available for the launching, landing and storage of aircraft on platforms. The type values are self-explanatory but the aircraft size requires a short explanation.

In the database, aircraft are given a size in the aircraft annex. In the game this is expressed as a numerical value. These values are:

- Small Aircraft = 2 points
- Medium Aircraft = 3
- Large Aircraft = 4
- Very Large Aircraft = 8

Small aircraft can use any facility. Aircraft can use any air facility providing it's as big as or bigger than it is size rating. Therefore:

One Very Large Aircraft Facility can hold two Large Aircraft, two and 2/3rds Medium Aircraft, and four Small Aircraft.

25.7. COMMS

As you can imagine, the Comms annex holds all the data that allows platforms and weapons to communicate among each other. This annex plays an important part in the DB because without comms, many weapons would not be able to work. SARH and Command Guided weapons wouldn't be able to work and ships would not be able to share data and there couldn't be an integrated defensive posture in the event of an attack.

All platforms and a large number of weapons will require communications units to talk and share information to one another. In H3ANW, in general communications are very abstracted. A particular aspect of communications that some players find undesirable is when the difficulty level is set to full realism and submerged submarines are not be able to communicate with any other platform. In game, this means you lose contact (and thus control over) that unit. If this sort of situation does not appeal to the player, the auto datalinks feature can be turned on to avoid this from happening. This is a trade off though because auto datalinks means you are in constant communication with your units since all platforms default to the longest range set on a platform in game. The value at the bottom of the unit display is the range of the comms unit in nautical miles.

Platforms, as a general rule require several comm units at the least. The types of comms in the game are as follows:

ELF Link – Extremely Low Frequency. This is a very slow communication method which can be received by submerged submarines. Only facilities and submarines use it

Type 75 Sonobuoy Link – This is the comm unit for Russian Type 75 LOFAR sonobuoys. It is used by those sonobuoys and by the aircraft that use them, namely Bears and May's and some ASW ships like the Udaloy and Kara.

BM Sonobuoy Link – The comm unit for the Russian BM series of buoys. It is an older system that is used by ships such as Kashin, Kanin, and Kresta classes.

RGB Sonobuoy Link – The current standard Russian sonobuoy system. There are many types in service and it's in widespread use on many Russian ships and aircraft.

French Sonobuoy Link – The French Navy have developed their own series of sonobuoys for use from their helicopters and MPAs.

NATO Sonobuoy Link – The universal western standard sonobuoy link, it is in widespread use throughout the globe. All NATO members and NATO aligned nations use this link.

Radio – The most widely used method of communication in the game.

Visual Comm – This is meant to indicate signal flags, signal lights, or signal mirrors. It cannot be jammed.

Laser Comm – A very new and very fast method of communication. Used mostly by ground forces.

Commercial SATCOM – A generic setting for long range communication sets. Widely used and easily hacked into.

A346Z Datalink – This is the datalink for aircraft that use the Big Bulge radar system. It is meant to pass on targeting info to strike aircraft. Bear and certain variants of Badger bombers use this to provide missile targeting for attacks.

Walleye Datalink – The datalink used by the Walleye glide bomb. It is also used in early versions of the AGM-84E SLAM missile.

GBU-15 Datalink – The datalink used by the GBU-15 EO guided bomb and the AGM-130 powered guided bomb.

APD-15 Datalink – The tactical datalink used by MiG-31 aircraft and possibly by Su-27 aircraft as well.

Missile Guidance Unit – Generic missile guidance comms.

LAMPS Datalink – The Light Airborne Multi-Purpose System is the current ship to helo datalink system used by the US Navy. It is used my most US surface ships and by H-2 and H-60 series helicopters.

Punch Bowl SATCOM – Dedicated missile targeting SATCOM system used by Russia on it's RORSATs, EORSATS, and the newer missile subs and cruisers such as Kirov, Oscar, Slava, etc.

Syracuse SATCOM – The standard satellite communications system used by the French Navy; it is exported to many nations.

TERMA Datalink – Data transfer system used mainly by the Danish Armed Forces. Used on board surface ships and as part of the Danish Coastal Defense system.

Link W – An unlicensed copy of the NATO Link 11 system. This system is used by France, Saudi Arabia and probably by China as well. It is associated with the TAVITAC and VEGA combat systems.

Link 4 – Datalink system used by France and the US for interceptor control; it is widely used on carriers and missile cruisers.

Link 10 – Tactical data system developed in the UK and used by Belgian, Dutch, Greek and Royal Navy vessels. It is broadly similar to but not compatible with Link 11.

Link 11 – The current NATO standard datalink system. It has been very widely exported and has many variants. It is carried on almost every NATO warship, MPA, and helicopter. There are also plans to fit all NATO (and Australian) subs with Link 11.

Link 16 – The next generation successor to Link 11, it is currently entering service in the USN on board combat aircraft and major surface combatants. Plans call for it to be exported to several nations allied with the US.

PEAB TDMA Datalink – The standard datalink system used by the Swedish Navy.

Big Ball SATCOM – Strategic communications system used by the Russian Armed Forces; it is similar to DSCS or FLTSATCOM.

Link Y – A variant of Link 10; it is very widely exported.

Link Z – A variant of Link 14 for use by non-NATO nations.

Skynet SATCOM – The current standard satellite communications system for the British Armed Forces. Thailand uses a variant called Star of Siam. *Will become self aware and attempt to eliminate mankind.*

DSCS SATCOM – Defense Satellite Communications System. This high-volume system is used by the US DoD and Diplomatic departments.

FLTSATCOM – The USN's high priority fleet wide communications system.

Have Quick – US Tri-Service radio system. Allegedly jam proof.

AKT-22 Datalink – The datalink system used by LAMPS equipped helicopters.

MILSTAR SATCOM – The new Tri-Service military communications system for the US Armed Forces.

SSIXS SATCOM – Submarine Satellite Information Exchange. Similar to the Punch Bowl SATCOM system except it is exclusively used by submarines.

Link 14 – A datalink system developed for ships that lack a combat system or a core central computer system. It is, in effect, a teletype version of Link 11.

One Way Wire Guidance – A guidance system where the weapon is guided by commands sent through it. ATGMs and early torpedoes use this.

Two Way Wire Guidance – A guidance system where sensor data is transmitted back to the launching platform. This in effect turns the weapon into a remote controlled autonomous vehicle.

Land Line – Communication via direct cable, fiber-optic, etc. Used by facilities.

25.7.1. COMMUNICATION DATA FLAGS

Broadcast – This indicates that the comm unit in question is capable of being jammed as its frequencies are widely known.

Secure – Used my military forces on discrete frequencies, these radios are considered jam-proof in the game.

Receive Only – This flag is used for certain datalink systems that operate as slaves to a central unit. The Russian Bell Spike system is one such example. The unit in question can only receive message and not transmit any information.

Send Only – This rarely used flag is for systems that broadcast data over a wide network. National defense systems such as the North Warning System send data back to NORAD facilities in the US and Canada for processing.

LOS Limited – Used by higher frequency radios, this signifies that the platform using this cannot transmit information unless it is in sight of another.

25.7.2. FREQUENCY BANDS

Code (Name)	Frequency Bands
ELF (Extremely Low Frequency)	Below 10 KHz
VLF (Very Low Frequency)	10 KHz to 30 KHz
LF (Low Frequency)	30 KHz to 300 KHz
MF (Medium Frequency)	300 KHz to 3 MHz
HF (High Frequency)	3 MHz to 30 MHz
VHF (Very High Frequency)	30 MHz to 328.6 MHz

Code (Name)	Frequency Bands
UHF (Ultra High Frequency)	328.6 MHz to 2.9 GHz
SHF (Super High Frequency)	2.9 GHz to 30 GHz
EHF (Extremely High Frequency)	Above 30 GHz (not supported in game)

25.8. FUEL

The simplest of annexes, this one stores all the information for fuel only. There are the following types of fuels:

Air Independent – This is basically liquid oxygen for use in AIP engines to allow submarines to spend extended time underwater without snorkeling.

Aviation Fuel – Any fuel used by aircraft.

Battery – Units of Battery remaining. Mainly used by SSKs when underwater.

Diesel Fuel – The most common type of fuel, mainly used in ships.

Gas Fuel – This is the type of fuel used by gas turbine powered vessels (e.g. LM-2500).

Oil Fuel – Used for steam powered vessels, this is also known as Bunker C.

Rocket Fuel – Any fuel that will power a rocket, ramjet, or any other non-air breathing powered weapon.

Torpedo Fuel – Any type of fuel used by Torpedoes.

Weapon Coast Time – Used for gliding weapons such as the JSOW or TALD, it calculates the amount of glide time the weapon will have.

25.9. LOADOUTS

The Loadout Annex stores all the possible combinations of weapons that can be carried by aircraft in that DB. It takes weapon records from their annex and applies them in a listed format.

25.9.1. LOADOUT COMPONENTS

ID# - The numerical listing of the loadout for the purpose of being read by the game engine.

Name – The name applied to a loadout.

There are two naming systems in use right now. The system used by HUD3 editors and the original TSPI system. The HUD3 system uses a 6 number system to keep track of the loadouts, their function and the type of plane they are used on. The original TSPI system was easier to read but less descriptive with only loadout type and the main weapon it carries. The original TSPI system uses abbreviations for its loadout description and a numerical count method.

Code	Loadout
OCA	Offensive Counter-Air – Any mission using anti-runway weapons.

Code	Loadout
ASW-DC	Anti-Submarine Warfare using Depth Charges/Bombs
ASW-T	Anti-Submarine Warfare using torpedoes (either guided or un-guided)
ATA	Air to Air using missiles and/or gun pods
ECM-P	Electronic Warfare using Sensor Pods (i.e. ALQ-99)
Ferry	Aircraft Ferry using Drop/Ferry Tanks (aircraft is usually un-armed)
IB	Iron Bombs such as Mk80 series and FAB series.
PGM-B	Precision Guided Munitions using Bombs (Guided/Smart Bombs)
PGM-M	Precision Guided Munitions using Missiles (i.e. Maverick, AS-14)
Recon	Reconnaissance using either pods or Mk1 eyeball
SEAD-D	Suppression of Enemy Air Defenses using Decoys (i.e. TALD)
SEAD-M	Suppression of Enemy Air Defenses using Missiles (i.e. HARM, AS-11)
SO	Stand-Off; any weapon that can be launched without danger to the aircraft (Harpoon, AS-15)
Tanker	Any aircraft equipped with a buddy store. Does not apply to dedicated tankers like the KC-10.
UG-C	Un-Guided Cluster weapons (Mk20, RBK series)
UG-R	Un-Guided Rocket weapons (HYDRA, S-80)

Other suggested load-outs:

Airborne	Troop simulator, from squad to company-sized units.
Bio	Biological dispersal system and storage, could be fertilizer, insecticide, defoliant, or similar.
Bucket	Forest fire style fighting bucket (helo only)
Cargo	Usually used on large aircraft to simulate shipping ammo between bases and ships.
ECM-M	Electronic Counter-Measures. Loadouts using a combination of EW pods and missiles.
Exercise	Loadouts using inert (warhead is a 1 DP flare) missiles.
FireSupr	Fire fighting suppressant

Airborne	Troop simulator, from squad to company-sized units.
Patrol	Combination loadouts, using a wide variety (usually 2 to 4) of weapons. Used on MPAs and Helicopters.
Relief	Humanitarian relief supplies
UG-G	Un-Guided Guns. Pod mounted guns on some aircraft and helicopters.
UG-I	Un-Guided Incendiary; this includes napalm and other fire gels.

ROF – ROF stands for 'Rate of Fire'. There are 2 ways of expressing it in the DB. A positive number means that the ROF is that many seconds. (i.e. an ROF of 15 means that the weapon in question will fire once every 15 seconds). A negative number indicates increments of 30 seconds. Therefore a ROF of -2 will mean the weapon will fire once every minute.

Capacity – This is the total weapons capacity of the loadout in question. There is no practical limit to this and the DB editor automatically calculates this when you place weapons records into loadouts.

Ready Time – This allows the editor to alter the amount of time it takes an aircraft to be readied for its next mission. This number is totally subjective and up to the editor. This is actually a fairly advanced setting because it can alter the speed and course of a scenario. Generally speaking, the larger the aircraft the longer it will take to make ready for the next mission. That being said however, tactical aircraft can be turned around very quickly indeed depending on the situation and the weapons involved.

Target type – This is an automatic setting that describes what targets the loadout is capable of attacking. This is a reference setting and doesn't apply to the game in any way.

25.10. MAGAZINES

This annex holds all spare weapons that a ship, sub, or facility will carry for later use. Once a mount on one of these platforms has expended all its weapons, it will automatically reload from the available magazines. Facility magazines currently do not work as of this writing and special exceptions have to be made for them to have sustainability in a scenario.

The Magazine annex is very simple to use. Simply select the magazine you wish to modify and select the weapon record that you wish to add to that magazine from the pull down menu. Duplicate weapons records are indicated by the Quantity field next to the pull down selection for the weapon record.

Other fields in the Magazine annex are:

Armor – This is an indication of how protected the magazine is from weapons hits against a platform. It is similar to the armor levels mentioned earlier.

ROF – The Rate of Fire for the Magazine is its ROF to move weapons from the magazine to the mount in question. Values are the same as in the Loadouts

Capacity – The maximum amount of weapons that the magazine can hold. It cannot exceed 65000.

Version – Number separating different versions. It is of no consequence to game or DB operation, but helps keep things straight for scenario designers.

UnRep Flag – This removes the magazine from the sanity check feature of H3DC. The sanity check would look for weapons in magazines that had no mount that could fire them. This would lead to a very large check file for aircraft carriers because the AC ordinance load isn't taken into account.

25.11. MOUNTS

The Mount annex, used by all platform types, contains the readily available weapons for the assigned platform. Mounts operate similarly to the Magazine Annex.

The **Sensor field** is used for mounts that have on board sensors. Examples are point defense weapons like Phalanx or mobile AAA vehicles such as the 2S6 Tunguska. It is important to know that when a mount has an on board sensor that the weapon on the mount be slaved to that sensor. This will be described later in the weapon annex description. The mount must be set to auto.

The sensor and weapons fields operate similarly to the Magazine weapons field in that annex.

The rest of the annex is similar to the Magazine annex with one exception. There is an 'auto' flag for all mounts. When selected the mount will operate independently from the human player and will attack every target it can given its weapons parameters.

25.12. PROPULSION

In H3ANW, all platforms and most weapons require propulsion of some sort for obvious reasons. Failure to have an engine of some kind in the required platforms or weapons will result in a non-functioning (unable to move) platform or weapon, which may cause unwanted game behavior...

There are five entry fields and two more sub-fields in the Propulsion annex.

ID – This is the numerical ID number assigned to the propulsion unit by the editor.

Name – The name assigned to the propulsion unit….at your discretion

Primary Type – The main type of propulsion. In most cases this will be the only type of propulsion used. Merchant ships, aircraft, and submarines will only have primary propulsion.

Boost Type – The secondary type of propulsion for ships with combined propulsion systems, such as CODAG (Combined Diesel and Gas), CODAD (Combined Diesel and Diesel), and COGAG (Combined Gas and Gas); in the case of Combined Diesel and Gas systems (CODAG) the primary type would be diesel and the boost type would be gas turbine.

Version – A generic value to differentiate between different versions of the same entry.

Altitude/Speed – This field sets the altitude and speed characteristics of the engine and platform. For ships, this is moot because they operate on the surface and are always at zero altitude. Submarines always operate in negative altitudes (under water) while planes cannot go any lower than 10 meters and even then they require to be over the water with their terrain following flags set.

It is important to note that speed increase isn't logarithmic but rather inversely exponential. That is, the speed increase be less at each increasing level. This is especially important for ships and

submarines, as they require a tremendous increase in power for each extra knot of speed. A typical speed breakdown would be this:

Speed Setting	Example Values
Slow	5 knots
Half	16 knots (11 knot increase)
Full	24 knots (8 knot increase)
Flank	29 knots (5 knot increase)

The altitude bands are for aircraft and subs. They indicate the varying altitudes/depths that aircraft/subs use. This is a fairly easy concept for subs because they will retain the same speeds no matter what depth they are.

Important Note: The altitude settings in the Propulsion annex take priority over the Altitude settings in the Weapon annex. The settings in the Weapon annex are for launch parameters only. In-game users will have to abide by the settings in the Propulsion annex which are not visible in game. If you have problems launching weapons in game, then one of the first places to check is the Altitude settings in the Propulsion annex, as they are usually the cause of several issues in this regard.

25.12.1. ALTITUDE AND DEPTHS FROM THE HARPOON 4TH EDITION RULES

Alt Band	Range in Meters	Range in Feet
Very High	10501 to 20000 m	34450 to 65617 ft.
High	7501 to 10500 m	24607 to 34449 ft.
Medium	2001 to 7500 m	6563 to 24606 ft.
Low	101 to 2000 m	329 to 6562 ft.
Nap of the Earth	31 to 100 m	99 to 328 ft. (over land, risky for fixed wing aircraft)
Very Low	0 to 30 m	0 to 98 ft. (over land, risky for fixed wing aircraft)
Surface	0 m	0 ft.
Periscope/Snorkel	0 to 25 m	0 to 82 ft.
Shallow	26 to 50 m	83 to 164 ft.
Intermediate I	51 to 100 m	165 to 328 ft.
Intermediate II	101 to 200 m	329 to 656 ft.
Intermediate III	201 to 300 m	657 to 984 ft.

Alt Band	Range in Meters	Range in Feet
Intermediate IV	301 to 400 m	985 to 1312 ft.
Intermediate V	401 to 500 m	1313 to 1640 ft.
Deep I	501 to 600 m	1641 to 1968 ft.
Deep II	601 to 750 m	1969 to 2460 ft.
Deep III	751 to 900 m	2461 to 2952 ft.
Deep IV	901 to 1050 m	2953 to 3444 ft.
Deep V	1051 to 1200 m	3445 to 3936 ft.
Very Deep	1201+ m	3937+ ft.

For aircraft it's a little more complicated. Not only do aircraft operate at different altitude bands; they also have different fuel consumption rates at each different band. This can be a little difficult, as there is little information on this regarding modern aircraft, hence the use of the Harpoon 4th Edition data as a fallback.

25.13. FUEL CONSUMPTION

The second field is for fuel consumption for varying altitude bands. Calculating fuel consumption for a platform is the most complicated item of any database second to sensor creation.

25.13.1. CALCULATING SHIP AND DIESEL SUB BURN RATES IN H3ANW

This is a super advanced topic and to that end the Community has developed and has maintained a spreadsheet called the *Platform Assistant*. Copies are available via the AGSI Wiki (see http://wiki.computerharpoon.com).

25.13.2. SOME GUIDELINES FOR FUEL

1. Do not include Aviation Fuel as fuel load. The aircraft in H3ANW do not need it to be present to refuel on a ship and they are automatically refueled when they land.
2. On ships, some Gas Turbines can burn gas, diesel and aviation fuel types. This is especially relevant to co-engine ships. Diesel is best as all ship engines can burn it.
3. As the fuel load is not seen by the player, you don't necessarily have to have the exact load as specified by your text reference. What is important, is to get the Endurance shown correct.

25.13.3. CALCULATING ENDURANCE FOR SHIP ENGINES WITH NO CO-TYPE

The fuel fields will require some algebra to properly fill in. Note that only the top three fields of each column will be used as ships will only operate in Alt Band 1. Leave the Boost Propulsion blank.

Due to the complexity of this topic, you are referred to the AGSI Wiki http://wiki.computerharpoon.com and the H3 ANW Platform Assistant (found on the Wiki).

25.13.4. CALCULATING ENDURANCE FOR SHIP ENGINES WITH A CO-TYPE

The fuel fields will require even more algebra to properly fill in. Note that only the top three fields of each column will be used as ships will only operate in Alt Band 1. Note that both Primary and Boost propulsion will need to be selected.

1. Boost Burn rate = (co-burn rate - (cruise burn rate x engine factor)) x Scale Factor where engine factor = 0.68 diesel, 0.80 gas turbine, 0.87 steam
2. Using PFEDIT, round to the nearest 0.1 and divide by 10, and then enter the boost burn rate.
3. Using DBEDIT, round to the nearest 0.1 and multiply by 10, then enter the boost burn rate.

25.13.5. SUBMARINE POWERPLANTS

1. Submarines may have two engines and each engine may use two altitude bands.
2. Altitude Band 1 must always contain the altitude range for which the max speed of the sub is achieved. Max surface speed is usually slower than max dived / snorkel speed.
3. Electric engines require batteries. A battery fuel load is given in seconds of endurance at 5 knots.

Due to the complexity of this topic, you are referred to the AGSI Wiki http://wiki.computerharpoon.com and the H3 ANW Platform Assistant (found on the Wiki).

25.14. SENSORS

The most complicated and important part of H3ANW is its sensors. Without it, naval combat simply isn't possible. Sensors permeate every aspect of platforms and weapons and provide the player with situational awareness of the battlefield.

There are 17 classes of sensors in the annex:

Radar – (RAdio Detection And Ranging) The most common type of sensor, radar is also the oldest sensor in operation. Only the searchlight has been around longer.

Visual – The visual eye. On board all platforms.

Infrared – A passive sensor, this is a visual form sensor that can see into the infrared part of the spectrum that isn't visible to the eye.

Passive Sonar – (SOund Navigation And Ranging) A listening sonar that is carried by most ships frigate sized and larger. It is totally passive with some systems having extreme range.

Active/Passive Sonar – A dual mode sonar usually carried by subs and anti-submarine ships, they have the ability to send out sound pulses underwater that function in a manner similar to radar.

ESM – (Electronic Surveillance Measures) A passive system that listens for radar (ESM) and communication signals (COMINT/SIGINT). This is mounted on almost all platforms. COMINT/SIGINT is not modeled in H3ANW.

ECM – (Electronic Counter Measures) Active jamming systems that can be used to block a variety of sensors, the most common of them being radar and communications jammers. However there are models that can jam light spectrums

Semi-Active – Any kind of sensor that relies on detecting the reflected emission of a specific (usually related) emitter for guidance. Mainly used by weapons seekers; semi-active radar homing is the most common type.

MAD – (Magnetic Anomaly Detection) An ASW sensor with very limited range, it detects the magnetic field of a submerged submarine. It cannot detect subs that have non-steel (i.e. Titanium) hulls.

Radar Suite – A system of radars that, share information using a Combat Direction System. It is not supported in H3ANW.

Sonar Suite – A system of sonar's that share information using a Combat Direction System. It is not supported in H3ANW.

Ranging Sonar – A short-range sonar system generally used for depth control and for bottom mapping. The hydrographic features of this sonar are not available in H3ANW.

Active VDS – (Variable Depth Sonar) In essence, an active sonar at the end of a long cable. This is a specialized type of sonar carried by many ships that operate in adverse sonar conditions.

Passive VDS – A passive sonar at the end of a long cable. Many ships that operate in adverse sonar conditions use this. They are different from towed arrays in that you can vary the depth of the VDS.

Towed Array Sonar – The most sensitive type of sonar available, this is a gang of passive sonar arrays at the end of a very long (300m or more) cable. A powerful sensor with some of them having tremendous ranges, they have no active capability what so ever. This is an excellent long range search system.

Active Sonar – Active sonar sends pulses under water in a manner similar to radar. They provide accurate fixes on targets but have much less range when compared to passive sonar.

Laser Designator – A difficult to detect version of Semi-Active guidance; primarily used for guided weapons, the laser designator shines a 'spot', generally not visible to the human eye to provide a target point for semi-active sensors to home in on.

25.14.1. SENSOR ANNEX FIELDS

Search Output – Used by radar's, active sonars and laser designators. This is the amount of power that is transmitted by the sensor in search mode.

Search Input – Used by radar's and active sonars. This is the value assigned to sensors to determine its sensitivity and therefore it's ability to pick up targets in search mode.

Track Output – Used by radar's, active sonar's and laser designators. This is the amount of power that is transmitted by the sensor in track mode.

Track Input – Used by radar's and active sonars. This is the value assigned to sensors to determine its sensitivity and therefore it's ability to pick up targets in track mode.

Passive Input – Used by passive sonars, visual, IR, and semi-active sensors. This is the sensitivity of the sensor and is similar to Search/Track Input.

Range (Minimum and Maximum) – The fixed minimum and maximum range, in nautical miles, for the sensor in question.

The Input/Output (I/O) values determine the sensitivity and thus the detection range of a sensor vs. a specific target. The rage range value is the maximum instrumented range of that sensor. If the I/O values are set for a really high sensitivity then the detection range will be cut off by the Range value. If the I/O values are set for a low sensitivity then the range value is moot as it probably won't detect anything out that far. Inputs are for all sensors (except designators), and Outputs are for active sensors

Altitude (Minimum and Maximum) – The fixed minimum and maximum altitude, in meters, for the sensor in question.

Range Accuracy – The accuracy of a sensor with deviance measured in nautical miles.

Angle Accuracy – The accuracy of a sensor with deviance measured in degrees.

Max Intercept – Applies to fire control sensors, this value indicates the number of channels available or targets that can be engaged at any one time.

25.14.2. SENSOR DATA FLAGS

Capabilities – This determines the capability of the sensor; whether it can search the surface, air, underwater (sonar), and a variety of other features.

Important Note: Surface Search radars in the game have the ability to do limited air search as well, therefore we have allowed surface search radars to have the ability to cover the VLOW and LOW altitude bands. This corresponds to real life experience with surface search radar where the operator (in this case me) has picked up low flying aircraft (including Tornados, Jaguars, F-16s, and F-4s) as well as wildlife (in my case - ducks flying over the water).

Search Frequency – The frequency that a sensor operates at in search mode; uses the NATO band naming system.

Track Frequency – The frequency that a sensor operates at in track mode; uses the NATO band naming system.

ECM Type – Used for ECM type sensors only, it indicates the type of radars that can be affected by the jammer. This feature is being upgraded with each new release.

Text – A brief text description of the sensor being edited.

25.15. WEAPONS

Weapons are the the tools of air/naval warfare. Most of them are designed to be launched from a mount, move to their target, localize the target, and attack it with an explosion. However, some are simply logistical aids (i.e. Drop Tanks), others carry sensors (i.e. Sono Bouys), and yet others contain electronic warfare aids (i.e. decoys).

25.15.1. WEAPON TYPES

Missile – Any guided weapon that has a self-sustaining motor and a guidance package.

Bomb – A dumb weapon that is dropped from an aircraft with a steel casing and explosive filler. An aircraft only weapon.

Rocket – A battlefield support weapon that is mainly a short-range missile with no guidance; mainly fired from aircraft and a few ships.

Gun – A ballistic projectile that may or may not be unguided. Found on ships and aircraft.

Torpedo – An underwater missile. The primary weapon of submarines, they are second only to missiles in their potency. Mk48, Spearfish, USET-95 are all torpedoes.

Depth Charge – An underwater bomb only effective when detonated in close proximity to a target submarine, examples are the Mk54, Mk11, and B-1 depth charges.

ASW Contact – A form of smaller depth charge that can be aimed to a limited extent and are fired in a multiple pattern of charges at the area of a target; a hit actually requires contact with the target for detonation to occur. Examples are the Russian RBU series and the older Mk10 Hedgehog.

Sensor Pod – Used on a variety of aircraft, these pods can be used for jamming, reconnaissance, or ELINT.

Sonobuoy – A small cylinder, dropped from aircraft that contains a sonar transducer and radio transmitter, designed to search for submarines. An example of a weapon that has a sensor warhead.

Drop Tank – A tank used to carry extra fuel aircraft.

Ferry Tank – A larger version of a drop tank, often carried internally.

Decoy – Any weapon meant to draw away an incoming missile from its real target. Affects guided weapons only.

Mine – A weapon that waits for a ship to strike or come into close proximity with it. A very cheap and powerful weapon that, is as yet to be implemented in H3ANW.

Buddy Store – A pod carrying hose, reel and drogue gear, used to re-fuel other aircraft in flight.

25.15.2. WEAPON ANNEX FIELDS

Length – The length of the weapon in meters or decimeters (one-tenth of a meter) if using the in-game editor.

Span – The diameter of the weapon in meters or decimeters if using the in game editor, sometimes wingspan is substituted for this depending on the weapon used.

Weight – The mass of the weapon in kilograms.

Waypoints – The ability of a weapon to undertake multiple waypoints and thus confuse the defender as to what direction the weapon was launched from. Not supported.

Climb Rate – The rate at which this weapon can change altitude, in meters per second; only applicable to missile and torpedo type weapons.

Launch Altitude – The minimum and maximum altitudes, in meters or 10's of meters if using the in game editor, that the weapon can be launched. If the launching platform is outside these perimeters then the weapon will not launch.

Launch Range – The minimum and maximum ranges, in nautical miles, that the weapons can be launched. If the launching platform is outside these perimeters then the weapon will not launch.

Launch Speed – The minimum and maximum speed, in knots, that a weapon can be launched at. If the launching platform is outside these perimeters then the weapon will not launch.

Cruise Altitude – The altitude, in meters, that a weapon will fly at when approaching the target. For this to work, the 'Level Cruise Flight' flag must be toggled.

Surface Near PH – The base percentage chance to hit for weapons versus surface targets. Used by guns and rockets at less than half range and by all other weapon always. Also used in the PK reduction value on decoys against acoustically homing weapons (i.e. torpedoes).

Surface Far PH – Base percentage chance to hit for guns and rockets at equal to or greater than half range.

Near ATA – This is a rating representing the ability of a weapon to hit an airborne target which is, in all likelihood, not cooperating. It is equal to the percentage chance to hit a non-maneuvering target minus 30. This rating is used by guns at less than half range, and by all other anti-air weapon at all ranges. Also used as a PK reduction value on decoys against radar homing weapons.

Far ATA – As above, but used by guns at half range or greater and IR decoys where Far ATA = PH reduction versus IR homing weapons. Also used as a PK reduction value on decoys against IR homing weapons

Separation – This field isn't used.

Terminal Trajectory – This field isn't used.

25.15.3. WEAPON DATA FLAGS

Illuminate at Launch – When this flag is activated, the weapon will require that a sensor track the target at launch time. This is used for weapons requiring target indication such as ASTER and CLAWS. Typically these missiles require target speed, heading and altitude so their own active sensors know where to look after launch.

Local Control Possible – This flag allows the user to control the weapon from the launching platform. Wire guided torpedoes and missiles usually have this flag. Supported for guns only.

No Diving Target Mod – With this flag, usually applied to advanced SAMs, will not have a PK penalty assigned to it when attacking a high speed diving target like the SS-N-12 or AS-4. SM-2 Standard, ASTER, and SA-N-9 are examples of weapons that will use this flag.

Level Cruise Flight – This flag indicates whether or not a weapon will descend (or climb) to a specific cursing altitude as set in the field above.

Limited to Mount Arc - Indicates that the weapon lacks the ability to radically change direction after launch. Such weapons are restricted to firing within their mount arc during Point Defense resolution.

ARM Loiter Capability – Applying to anti-radar missiles, this will have the weapon in question remain over the target area waiting for an appropriate target to appear, if one hasn't appeared already. Mind you what happens to the weapon while it loiters is another thing. Weapons can be shot down remember. It applies to ARMs like ALARM, and Tacit Rainbow.

ARM Tunable in Flight – Applies to ARMs that can be re-targeted while they are in flight; used only for the most advanced ARMs (none of which are in service in real life). Not supported as of v3.10.

Stern Chase AAM – Applying to the earliest of AAMs, weapons with this flag will only be able to launch against a target when its seeker can only see it's back end (engine exhaust). Weapons such as AIM-9A Sidewinder, AA-2 and PL-2 are covered under this flag.

Rear-Aspect AAM – Differing from the Stern Chase flag, these missiles can launch against a target providing it's in it's rear hemisphere (i.e. behind the 3-9 line). AA-8, R.550 Magic, and AIM-9J use this flag.

All-Aspect AAM – All-Aspect AAMs are able to attack their targets from any angle. However, this is still a launch envelope for the best PK for the missile. This flag is typical of third generation AAMs such as AIM-9M, MICA and PL-7.

Dogfight AAM – Fourth generation short range AAMs should have this flag toggled. They are all aspect weapons with high-G maneuverability provided by thrust vectoring. AIM-9X, AA-11 Archer, and Python 4 all should have this flag.

ARM Target Memory – ARMs with this flag will have an inertial guidance unit that allows it to 'track' a target based on it's last known position, course, and speed after it shuts down it's radar. AGM-88 HARM has a target memory.

Home On Jam – Some weapons have the ability to home in on the signals emitted by jamming aircraft. This mainly applies to AAMs such as AIM-54 Phoenix and early models of the AIM-120 AMRAAM.

Terminal Illumination – This flag requires the weapon to be guided by a radar during its terminal homing process. All semi-active homing missiles are required to have this flag. All Standard series SAMs, Sea Sparrow, SA-N-6, and Laser Guided Bombs all require terminal illumination.

Capable vs. Seaskimmer – Some weapons are not capable to depress enough or their guidance radars can't distinguish ultra-low level flying targets like missiles. When this flag is toggled then a weapon will be able to detect and engage low flying targets. CIWS systems, the more modern SAMs and guns will have this capability.

Search Pattern – Weapons with this flag will be able to carry out a search pattern using whatever sensor they carry if they do not find a target at their pre-planned activation point. When in search mode, they will adopt a snake search pattern until they acquire a target or run out of fuel. Weapons without this flag, if they don't acquire a target (or miss) will go ballistic and eventually run out of fuel.

Drive-Through Logic – This flag is used to indicate torpedoes that can classify decoys and drive around or through them. The Mk48 Mod 5 ADCAP, USET-95 and the Mu-90 Impact are

all capable of this. Any weapon created with this flag will only be effective if it also has sensor capable of searching.

Bearing-Only Launch – A common capability of modern weapons systems (and one of the more powerful tools of H3ANW) is the ability of a weapon to be launched, set to activate at a point designated by the user. Most smart weapons (missiles and torpedoes) have this capability.

Wake-Following Torpedo – Wake following torpedoes traverse a snake pursuit pattern. A sensor determines when it enters the disturbed water that makes up the ship's wake. Only a few weapons have this capability, the most popular example being the Russian Type 65 torpedo that was recently retired from Russian service.

Straight-Running Torpedo – This applies to older torpedoes such as the British Mk8 (used to sink the General Belgrano during the Falklands War). Weapons with this flag will run ballistic. They may or may not hit the target and are obsolete in the face of more modern weapons.

25.15.4. TARGETS

This for the most part is self explanatory. Targets can be ships, subs, aircraft, radars (which include the platform it resides on), Runways, and Structures.

Surface Vessel – Enables the weapon to target surface vessels (including surfaced submarines).

Submarine – Enables the weapon to target submerged submarines.

Aircraft - Enables the weapon to target aircraft.

Missile - Enables the weapon to target missiles.

Radar - Enables the weapon to target emitting radars and jammers. If the weapon has a Surface Near Ph, it is usable against ships, surfaced submarines, and land facilities. If the weapon has a Near ATA, it is usable against aircraft and missiles.

Runway - The weapon is usable against large slabs of concrete. Runways can NEVER be completely destroyed, but can be reduced to near-worthlessness.

Land Structure – Any semi-permanent or permanent building that is not armored, from Office buildings to guard shacks.

Hardened Structure – Any building that is protected with any kind of armor. Bunkers, pillboxes, generally reinforced or underground structures intended to resist attack.

Soft Target – Any unarmored mobile target such as trucks, cars, APCs (like the M113), etc.

Hardened Target – Armored non-structural targets (dug-in troops and guns, tanks).

Communications – Some weapons will require datalinks to communicate with their launching platforms. This usually involves the creation of a weapon datalink in the comm annex and placing it on the weapon. In all cases, there has to be 2 comm entries, one for the launching platform and another for the weapon. The launching platform unit must be set to send only while the weapon unit has to be set to receive only. The weapon will then trade information with the launching platform (and only the launching platform). This is a minor code issue that will be addressed in the future.

Directors – A large number of weapons require guidance or target information from off-board (from the weapon) sensors to hit their targets. This entry will have to be filled if the weapon in question has either the 'terminal illumination' or 'illuminate at launch' flags toggled. Entries

here are taken directly from the sensor annex using their ID numbers. A drop down menu is provided for it if you wish to search for the correct sensor.

This generally applies to semi-active homing weapons and beam riders but can apply to certain active homers as well. The ASTER SAM is one of these weapons. It has the 'illuminate at launch' flag toggled because it requires information from a fire control radar so that it can calculate its intercept. After launch the missile uses it's own autopilot (like in the Standard SAM) to guide it to a point where it's active seeker switches on. At this point the missile will attempt to intercept the target.

Engines – With the exception of sensor pods, sonobuoys, iron bombs (both guided and unguided) and rockets, all weapons require some form of propulsion. As explained in the propulsion and fuel annex, the type of propulsion will depend on the weapon. Generally, jets, rockets and gliders are the most common. It's important to note that as of v3.10 weapons have a constant fuel consumption rate regardless of range, altitude or launch point.

Fuel – As explained in the fuel annex above, this will store the fuel listing for the weapon in question. See the fuel section above for more information.

Sensors – A large number of weapons carry sensors to assist in hitting their targets. Dumb weapons, drop tanks, rockets and depth charges don't require sensors because they simply have to hit the ground or reach a specific depth before they detonate. Most weapons though have sensors. Weapon seekers can be radar, IR, or optical but they all must have the ability to see the target for the weapon to hit it. Sensors are selected directly from the sensor annex in the same manner as directors are selected. For a Sensor Pod, these sensors will be added to an aircraft equipped with a loadout including this weapon.

Next to the sensor list will be the sensor arc selection box. It will determine the field of view of the sensor. In most cases this should be the forward arc but there will be some cases where they will cover other arcs as well. Sensor pods will fall under this most often with ESM, ECM and SLAR pods being the most common.

Warhead – This is the actual payload of the weapon. It is taken directly from the warhead annex described above. While weapons can have multiple warheads, most do not.

Text – This is a brief text description of the weapon.

25.16. WEAPON RECORDS ANNEX

One of the simpler but more important annexes is the Weapon Records section. This annex tells the game the type and number of weapons that a mount, magazine, or aircraft loadout will contain. Without weapon records (or 'weaponrecs' as they are sometimes called) planes and ships won't be able to launch missiles and submarines won't be able to reload their torpedo tubes.

The Weapon Record annex contains the following entries:

ID Number – The ID number of the Weapon Record itself

Weapon ID Number – The ID number of the Weapon for that Weapon Record

Weapon – The name of the weapon indicated by its ID number.

Default Load – The initial number of weapons that the weapon record will carry for the mount, magazine, or loadout.

Max. Load – The maximum number of weapon that the weapon record can carry in a mount, magazine or loadout.

ROF – Rate of Fire for the weapon record. In all cases, the ROF should match the ROF of the mount, magazine or loadout. Loadouts generally have an ROF of one so this shouldn't be a big issue. However mounts have differing ROFs and care should be taken to make sure they match. There is a limitation in the game regarding mounts like the USN Mk26 twin rail. The Mk26 will pop out two missiles every 10 seconds. In this case you'd set the Mk26 ROF and its stored weapon records to five seconds each. This ensures the adequate VOLUME of fire if not the correct method of fire. Firing rates are slaved to the mount first and then to the weapon record second.

Multiple – The number of weapons in the record that will indicate a single 'salvo' has been launched. Once the number of weapons fired reaches the multiple value, then one slot (i.e. an empty slot) will open on the weapon record.

An example of this is the Evolved Sea Sparrow Missile. This SAM has been designed to be quad-packed in Mk41 VLS systems. What this means is that four ESSMs can be carried in a single VLS cell. This in turn means the weapon record multiple must be set to a value of 4. Once all 4 ESSMs in that cell have been fired then that cell is empty and ready to be reloaded.

Version – This is a simple version number to help prevent duplication. It has a generic value.

25.17. WARHEADS

The Warhead Annex, used only by the Weapons annex, contains all the payloads for all weapons in the database. The following fields are in the annex:

ID Number – The reference number used by the editor to keep track of the warheads.

Warhead Name – This is the name of the warhead.

Text Index – Warheads do not have Text Indices. (Only displayed in the in game editor)

Warhead Type – Despite its name, this annex doesn't just hold explosive warheads. There are a variety of types, listed as follows:

- **Enhanced Conventional –** Represents Fuel-Air explosives and other, more exotic, methods to improve the efficiency of conventional payloads. Due to their vastly increased yield, these weapons are modeled as area-attack weapons doing atmosphere-transmitted shock. (Check game support for this flag, text taken from H2 manual)
- **Sensor –** Meant to simulate the dispensed chaff, the value is actually the ID number of the sensor in the sensor annex. In Chaff rounds this is a low power ECM sensor. For flares it's a low power IRCM.
- **Weapon –** This is for weapons that contain another weapon as their primary warhead. The DP number in this case is the ID number of the weapon in the weapon annex. Weapons that use this type of warhead are ASROC, SS-N-16, where these weapons employ a "slave" rocket system to carry an ASW torpedo to optimal attack position. By clicking on the Weapon radio button or weapon

index number drop down menu, you should see a list of all items in the Weapon annex appear in the Payload Annex. When the weapon reaches its target, it will create a number of weapons of this type equal to the number in the Number field which will then be appropriately executed (some will have immediate effects, such as depth charges, while others will be released and conduct their own searches and attacks).

Warning! If a weapon's warhead has a weapon payload of the same type as the weapon upon which it is mounted, H3ANW will try to load the combat resolution information for the weapon, then notice that there is a warhead, and try to call the weapon which is the payload of the warhead, then notice that the weapon carries a warhead, etc; the game will probably lock up.

» **Depth Charge –** The Depth Charge warhead represents underwater explosive warheads designed to explode at a preset depth and damage targets by shock.

» **Fragmentation –** An explosive warhead designed to disperse small pelletized fragments these fragments have reduced effects against armor, but increased effects against units without armor such as personnel. Many types of cluster warhead deploy fragmentation warheads.. WARNING: In H3ANW, this warhead will destroy any target it hits no matter what the DP values of the target or warhead. This type of warhead should not be used except by weapons that are meant to be used against airborne targets ONLY. Any type of armor on the target will nullify the effects however.

» **Torpedo –** Torpedo warhead, operates much in the same way as a Depth Charge. Has twice the explosive power of a conventional bomb, due to the hydraulic effect of the surrounding water. This represents explosive charge payloads which are designed to detonate on contact with the target. Also represents such warhead types as Common, High Capacity, and Semi-Armor Piercing.

» **Armor Piercing –** Used by weapons that are used to employed against armored targets. Usually applies to guided bombs and anti-runway weapons.

» **HEAT –** High Explosive Anti-Tank ammunition; used against armored vehicles such as tanks. Examples of HEAT warheads are TOW, TRIGAT, and AT-16.

» **Fire –** (NAPALM, WP) A warhead which consists primarily of a combustive agent. These types of warhead do not inflict DP damage, but rather start fires. The Capacity of the warhead is calculated exactly as for standard warheads..

» **Nuclear Air Burst –** A nuclear device designed to detonate above the surface, destroying targets through a combination of EMP, thermal effects, and atmospheric shock.

» **Nuclear Surface Burst –** A nuclear device designed to detonate at or just below the surface, destroying targets primarily through shock transmitted through the water or ground.

» **Nuclear Sub Burst –** A nuclear device designed to detonate underwater and destroy targets through water-transmitted shock. Mostly deployed as anti-submarine weapons, but usable against surface targets as well.

Gun Class - For gun and rocket warheads only, this field selects the effective diameter of the payload for armor penetration purposes. Select the entry corresponding to the diameter of the projectile (may be smaller than the bore diameter for sub-caliber or payload rounds) or warhead (may differ from rocket size).

Capacity - If the warhead type is set to weapon, this field will determine how many weapons to create when the 'warhead' is deployed.

Cluster - The warhead is a dispenser of smaller charges. Cluster warheads should have their damage determined normally based on the total mass of the sub-munitions. Cluster weapons attack a small area around the designated target.

25.17.1. DAMAGE POINT CALCULATION

As a general rule, all DP calculations are based on the weight of the **warhead** in kilograms. Therefore a Mk84 2000lb bomb has a DP value of 91, with 2000lbs = 909.1 kg, half of that being the HE of the warhead. That's 454.5 kg for a total of 91 DP. You round up the value.

Nuclear warheads express their DP value in kilo tonnes and tonnes. A negative value will be evaluated in tonnes and a positive value will be considered in kilo tonnes. Therefore a 400kT MIRV would have a DP value of 400.

25.17.2. EFFECT RADIUS

The Damage Radius field represents the area effected by the weapon's detonation. It is expressed in terms of nautical miles. The effective distance is modified for "well built" units. Submarines and ships with the shock resistant flag will reduce the damage and kill radii by one third as it applies to them. Hardened facilities will have the damage and kill radii reduced by one half as it applies to them.

If a damage radius is not set, nuclear warheads will have their base damage radius using the following equation. 1300*3√kt Yield.

25.18. TEXT ANNEX

This annex holds all the text descriptions in the DB. It is one of the lesser used annexes because it doesn't alter the .dat files but the .res files, which reside in another directory.

Altering text is very simple. Simply type whatever you want in the proper field and save the annex. Go to the platform in question and select the proper text ID number. There is no way to edit the .res files using the in game editor. If you wish to edit these files you will need to use the provided, but unsupported, RezEdit utility (see the Tools directory).

Proper use of this annex is very important to the game. Should the H3.res files have different text ID numbers than the values in the platform annexes then you will most

likely suffer a fatal game crash. It is important to remember to export the .res files to their own directory with the correct DB.

The location of the .res files can be found in your Harpoon3.ini file. The simplest solution to this would be to point a separate directory for the .res files. However this should only be done by experienced users. Novice users are better off leaving this alone for the time being. The only way to get a feel for this will be by trial and error. It's better for you to get experience understanding how the editor and file structure works before attempting this.

25.19. HARPOON COMPONENT FILES (HCF) – H3RE ONLY

Harpoon Component Files (HCF) allow the user to transfer either platform or components (either singly or multiple items) in a compressed file. The information is then exported to user named file with an HCF extension. The file then can be passed on to others editors using the import command described below. The files are very small (usually less than 15 kb) for easy transport.

During import of an HCF file the editor will also check (it will automatically ask the user if he/she wishes to) for duplication. If any duplicates are found, the editor will give the user the choice of either installing the duplicate item, or to use the original item or something similar to it.

This is a concept similar to the validation report described below. It also requires a fair amount of watching so that you don't reject items that you feel you need, even though they might be similar to another item in your DB.

26. H3ANW DATABASE COMPLIANCE

"The difference between a good and great officer is about ten seconds." - Admiral Arleigh Burke

26.1. H3ANW DB COMPLIANCE NOTES

This section details all changes that have been made to H3ANW Databases for v3.10

26.2. MISSILE SEEKERS

Missile seekers now perform more to real life than in previous versions. The determining factor here is the BOL flag. An active seeking weapon without the BOL flag will be considered to be a Homing All The Way type weapon (i.e. active as soon as it leaves the tube/rail) with the appropriate consequences if there is a friendly between the launched weapon and it's target. Weapons with the BOL flag will be considered capable of traveling some distance before requiring guidance (either from an FCR or it's own seeker).

For DB authors this means that all active homing weapons (Harpoon, AMRAAM, etc) will require the BOL flag to be present. Passive homing weapons that use ESM, EO and IR guidance will not require the BOL flag, however the seeker on the weapon to have to be able to see the target before that weapon will fire. This is analogous to getting the 'tone' prior to firing a sidewinder missile. The DB author must ensure that the seeker can see at least as far as the range of the weapon for the weapon to be able to use it's full range capacity. Some of you may remember that during Desert Storm A-10 pilots used the seeker head of the Maverick missile as a primitive

FLIR. Given the correct settings for the Maverick seeker then you should be able to do that as well.

Initial versions of the ANWDB feature an increased seeker head sensitivity to a very high level (I use –1500) and set the seekers max range to 1.5 times the maximum range of the weapon. This gives the seeker (and the plane) the time to pick out targets. It's up to you as DB author what input value you use however.

This feature must be applied over the entire DB, otherwise seekers will not perform properly.

26.3. UNDERWAY REPLENISHMENT

For Database authors, tankers will require correct fuel loads based on reference material. The number of ships that can be refueled at once is dependant on data flags and can be set to up to 4 per side.

26.3.1. SWITCHOLOGY

Underway refueling is accomplished by selecting the ship requiring fuel and hitting the logistics button [F5]. The cursor will then change to a crosshair and you move this to the oiler that you're going to use and double click on it. The thirsty platform will then lay in an intercept course for the oiler. Upon intercept, the thirsty ship will then enter into formation with the oiler and commence hook up.

Because many ships can also transfer stores (i.e. weapons) as well as fuel, the logistic transfer menu will come up first. Hit OK and there will be a notification in the message window that the ship is getting into position to refuel.

Hook up will take approximately 20 minutes of game time, give or take. Fuel transfer will then start and you'll get a message as to how much is being taken aboard.

Fuel transfer rate is determined by the field set in the DB editor. (for the older players this was the Repair Capacity field). For reference purposes, it should be noted that *Kaiser*-class oilers have eight cargo pumps with combined output of just under 5500 tons per hour. This works out to an average of 686 tons per pump per hour or a rate of 11.5 tons per minute per pump. This means that a Spruance class destroyer with 1500 tons of fuel capacity should be topped up by a single pump in a little over 2 hours. As a comparison, all RN ships use a standardized system that employs 6 inch hoses that transfer at a rate of 500 tons per hour per station. An Invincible class can be topped up in about 2 hours and frigates in about 40 minutes (from Combat Fleets).

There is a field in the ship annex of the DB editor to specify the amount of fuel transferred in tons per minute.

Underway Replenishment of Weapons functions exactly the same as fueling does. Transfer starts immediately and it's rate is determined by the ROF settings in the magazines of the UnRep ship.

Implementation of this is optional at the Database Author's discretion. Most H3ANW scenarios will never use this capability and only the most committed and anal scenario designers would ever make the time and effort to create a multi-week scenario that requires ships visit a UnRep group. That being said there are some scenario ideas that could make use of this. In any case this broadens the scope of the game into campaign territory.

26.4. MISSILE DEFENSE RATING (MDR)

The Missile Defense Rating is the number of weapons that will be allocated by the AI (or by staff if that is so checked) against a target. However, the target has to be ID'd for the MDR to be applied by the game engine. In the event that the specified target is un-identified the GE will fall back on a default allocation that is size dependant. These sizes are as follows:

Default MDR and Weapon Allocation

Ship Size	Default DP Value	Default Weapon Allocation
Very Large Ship	499+ DP	500 DP + 4 additional weapons
Large Ship	139 to 498 DP	250 DP + 4 additional weapons
Medium Ship	49 to 138 DP	120 DP + 2 additional weapons
Small Ship	9 to 38 DP	80 DP + 1 additional weapons
Very Small Ship	Less than 9 DP	20 DP + 1 additional weapons

Ultimately though, the number allocated will be wrong. This is the nature of incomplete information. Call it the fog of war.

Let's say that a flight of four A-6 Intruders carrying Harpoon missiles (45dp warheads) detects an un-identified large ship target. Should the player decide to attack the target while it is still un-identified then, the GE will allocate 10 of those missiles [250dp = 6 x 45dp warheads each + 4 extra missiles]. Best practice assigns a MDR based on the number of fire control channels that a ship has but … that is highly subjective. There are many other factors that determine the MDR for a vessel, such as DECM, decoys, and CIWSs.

Sample MDR formula:

Area Defense Systems (10 nm or greater anti-air range) MDR:

Number of shots vs. a mach 1 closing target x number of simultaneous engagements (director channels or launcher ROF) x the PK of the incoming weapon (0.75 to 0.85 depending on generation)

Point Defense Systems (<10nm anti-air range) MDR:

Number of firing arcs / 8

Distraction Systems

+1 if Defensive ECM system present

+1 if Chaff/Flare launcher present

This feature will be required for most ships but there is some leeway given in the fact that some platforms (i.e. small craft like a Boghammar) don't really require a Harpoon to destroy. Obviously the wide amount of variation between DBs will mean that the default value calculated will be different. The information I've supplied here is based on the ANW DB which uses the H4.x Master Rules so keep in mind that other DBs may produce different values based on warhead DP values.

26.5. ELECTRONIC WARFARE

Electronic Warfare in H3ANW is currently of two types, OECM (barrage) and DECM (point defense). For the purposes of the database, OECM will depend on the value in the search output value of the jammer in the sensor annex. DECM will use the track output value of the jammer in the sensor annex.

Burn through values for OECM will be estimated by use the H3ANW Platform Assistant (see AGSI Wiki ANW 3.10 section). It will be dependent on the output value of the jammer vs. the input and output values of the radar with the cross section of the jamming platform to add some flavor.

DECM values represent a reduction in PK as determined by the value set in the track output value. For example, an ALQ-131 ECM, considered a 2nd generation jamming pod will have a reduction in PK of 10%.

It is up to the database author what they consider to be an acceptable value for the PK reduction.

This feature must be applied over the entire DB. Typical PK reduction values will vary between 5 and 20% based on the H4.x Master Rules. Some systems have higher PKs for their DECM value but under no circumstances should the DECM value be negative. A negative value will produce a very high PK reduction with unrealistic results. We've discovered that a negative value leads to a 100% reduction in the PK value leading to the minimum PK value of 5% being applied to any weapon attacking a unit with a negative DECM value. This applies to decoys as well.

26.6. ACOUSTIC INTERCEPT AND RANGING (AIR)

AIR is basically ESM for submarines. In H3ANW it is dependent on sonar frequency. As per the H4.x Master Rules, passive sonars will hear active sonars of their own or one higher frequency band. Since range attenuation is dependent on the frequency of the emitting sonar and since helo and sonobuoy sonars are mostly high frequency they will be totally inaudible to a Low Frequency (LF) sonar. LF sonar have truly miserable range characteristics (less than direct path, which is 10nm or so). We don't have an explicit AIR model in the game yet, but you could use a Medium Frequency (MF) passive set with fairly poor performance as a fake AIR (except on older Soviet subs, which had MF sets with poor performance as their main search set already.

Using multiple frequencies won't work as the sensor will default to the lowest frequency.

This is an optional feature for DBs as many passive sonars will be able to pick up LF/MF sonars or MF/HF sonars. If you're really committed then you can create 2 separate AIR sets that can pick up all frequencies.

26.7. DIESEL ELECTRIC (DE) SUB FUEL BURN RATES

H3ANW features the battery discharge/recharge rates from the Harpoon 4.1 rules. The changes are:

- AIP no longer recharges. Initial implementation assumed onboard oxygen liquefaction which apparently is not done.
- Base recharge rate is 20x. Halved if at full or flank throttle. Halved if batteries above 70% of maximum.

- When charging, the engine(s) will operate at flank throttle, but the sub will move at the speed for the set throttle. This will appear anomalous in the unit report window, but the unit is moving at the stated speed, and the engines are in fact burning fuel at the stated throttle setting.
- AIP systems in the computer version are implemented as diesels with an independent oxygen supply (AIP Fuel) that burns at the same rate as the fuel. AIP systems consume fuel at the base rate for Idle/Creep, 2x for Cruise, and 3x for higher. At Idle/Creep they will recharge batteries at 2x, 1x at Cruise and not at all for higher.
- Speed effects on discharge rate. Discharge rate is multiplied by the base rate defined in the propulsion system (normally should be 1 for electric engines for fuel in seconds at cruise). From the Battery Discharge Table with 4 differences:
- At 0kt discharge rate is .5x
- At 19 knots discharge rate is 32.5x (interpolated, paper rate was 39 and pretty obviously incorrect)
- above 25kt discharge rate is +10x per knot to a max of 640
- Rates are interpolated in the cases where multiple speeds shared a paper rate.

There must be no propulsion gap for submarine engines. Many DBs have the electric motors' top altitude set at 30m. This will now cause an engine shutoff when going between periscope and shallow. Submarine electric motors should go all the way up to 0m (i.e. the surface). This change is due to the new hotkey ';' that toggles battery charge.

Since this affects all submarines this will be a required change for all DBs as the change in fuel consumption rates means that all DBs need to have corrected fuel loads. In the future, a new version of the H3ANW Platform Assistant will take this into account.

26.8. ANTI-SHIP PATROL MISSION BEHAVIOR

There's been a change to the behavior of ships assigned to surface patrol missions. They will operate at cruise speed until they detect a target and then they will alter course and speed to intercept. Once the target is ID'd to its satisfaction then the patrolling ship will resume its original course and speed. It will play catch up.

This affects scenarios. The user is reminded that they will have to re-consider the positioning of units in scenarios so that the AI doesn't use its new found speed advantage to unbalance the game.

26.9. AIRCRAFT FUEL BURN RATES

It should be noted that the boost value for engines reflects the multiplier for Full Military Power and NOT Afterburner. The afterburner modifier is hard coded to be 5x that of the boost value. According to the H4.1 x Master Rules, Turbo-Fan (TF) type jets (i.e. F-15) have a boost value of 4 and Turbo-Jet (TJ) type jets (i.e. A-6) have a boost value of 3.5. All other aircraft engines have a boost rate of 1.5 as per H4.1x Master Rules. The afterburner modifier is still present but

since only TF and TJ type jet engines have that capability then to add that extra value would deviate from realism.

Jet propulsion has been split into Turbojet and Turbofan categories. All previous Jet systems are now classed as Turbojets. The only difference is in the multiple applied to the boost rating when in afterburner. Turbojets are x3, Turbofans are x6. Full Power multiples are still adjustable, although the H4.1 Rule x Master Rules values should be used in the absence of specific information. The primary reason for the different full and afterburner multiples between turbojets and turbofans is that turbofans are considerably more efficient at subsonic speeds, and should have rather lower base fuel consumption for comparable performance. At transonic and supersonic speeds, much of this advantage is lost and the turbofan will have similar speed/range characteristics to a turbojet.

26.9.1. AIRCRAFT ALTITUDES BANDS

- VLow 0 - 30m (over Water Only)
- Nap Of the Earth (NOE) 30 - 100m (over Land Only)
- Low 101 - 2000m
- Med 2001 - 7500m
- High 7501 - 15240m
- VHigh 10501 - 25000m

Aircraft use the correct minimum altitude for the environment. Over sea, the Low altitude band extends down to 31m. Aircraft will not go below 31m over sea (Hazardous low-altitude flight is not yet implemented). Over land, the minimum altitude for aircraft is dependent on their capability. Helicopters and aircraft with Terrain-Following or Terrain Avoidance will operate down to 31m (the bottom of NOE). Other aircraft will be limited to a minimum of 101m. This altitude is relative to the ground, not sea level.

This is an optional feature based on the realism requirement of the DB author. However keep in mind that higher boost values will lead to an un-realistic reduction in fuel burn ranges so it's probably best to play with the values until you reach something you're happy with.

26.10. MISSILE FLIGHT TRAJECTORIES

Terminal dives are now modeled. DB authors are advised that the dive will begin when the seeker has picked up a target (i.e. at the Pre-Planned Activation Point). It will then follow a flight path that is based on the direct path of the flight geometry. Think a right triangle. This will allow databases to be able to accurately model flight altitudes of the various high altitude ASMs such as AS-4.

This is a low priority change as most DBs are currently set up to allow most weapons to engage high flyers at altitude by setting looser altitude launch parameters for defensive weapons. Keep in mind that should this be implemented it will drastically reduce defensive capabilities of the targeted ship because engagement will not take place until the weapon enters the engagement envelope.

26.11. ANTI-RADIATION MISSILES (ARM)

ARMs will now place their critical hit on the radar that they were targeted on. Warheads must be fragmentation to accomplish the correct critical hit. Fragmentation warheads will also no longer penetrate armor in any way, even the smallest value will deflect the warhead fragments. The sensor will still be destroyed however. As a side note, all air bursting weapons will now cause at least one critical hit.

This should be applied to DBs but the real change here is in warhead performance. Fragmentation warheads will not cause more realistic damage. Keep in mind that the seeker must still be able to see the target as described above in the section on seekers. Without this mod, ARMs will act more like ASMs and may or may not destroy the radar it was assigned to.

26.12. TORPEDO DEFENSE

Torpedoes are now marked as valid targets (in addition to ships, planes, subs, missiles, etc) to provide for anti-torpedo defense. The H3RE and the in H3DC have been modified to take this into account. DB authors are to make sure that the associated detection sensors are present or that existing sensors on the platform can detect the incoming torpedo enough for a shot. Weapon performance will follow the H4.x Master Rules in that anti-torpedo weapons will have a fixed PK of 30% against wake following and straight running torpedoes and 20% against acoustic homing weapons.

This is another long term project for DBs. However future systems like the US Navy's LCS and DDX will use active torpedo defense in their designs so DBs that use futuristic or hypothetical platforms (such as the ADB for example) should have this as an integral part of any defensive systems.

26.13. DECOYS

Decoys come in two flavors, active decoys (meant to simulate a target) and point defense decoys (PDD) (used during attack resolution). MOSS and TALD are examples of active decoys. SRBOC and Sea Gnat are examples of PDDs. For a decoy to simulate a target type, you will have to add an engine and fuel and associated jammers and sensors (if you so desire). The Decoy target type will specify the type of target to be simulated by the decoy. Therefore if you select a sub then it will show up as such on the other players screen. Keep in mind to have the appropriate propulsion for the decoys.

This might also be applied to long range chaff and active decoys like Nulka, but we haven't tested it extensively. It should work in theory, however.

This enhancement is related to the Defense ECM value as the simulation uses the PK value of the decoy as the reduction factor to apply to the attacking weapon. This value cannot be negative either. This change will dramatically affect multi-player games once users understand how the concept works. Incoming missiles will be flooded with a large number of fake contacts that have been Bearing Only Launched from targeted ships. Try to keep the number of decoys down to a manageable level, as the decoy launchers have a very high reload rate, perhaps 15 or more minutes, and each adds to the CPU and communications loads on the player's computer.

26.14. WEAPON SALVO ALLOCATION AND PROBABILITY OF KILL (PK)

The default allocation has been adjusted to be more pessimistic. Previously it was only firing 1 missile if it had an estimated Pk of 80%. It will now fire 2 weapons at planes and missiles unless the estimated Pk is 90% or higher.

This isn't really a DB change but rather a change in doctrine. It's placed here so that DB authors can change the PKs of their weapons to reflect different weapon allocation if they so wish.

26.15. EXPANDED COMM FLAGS

There are now comm flags for adjusting the channel usage of sonobuoys. Directional sonobuoys use multiple channels to transmit additional information to their parent unit. These flags are not currently implemented; they have been added to support future sonobuoy developments, as channel usage is a big limiting factor in the real world.

26.16. REBUILD ALL YOUR SCENARIOS

After you've made your DB changes, make sure to rebuild all your scenarios. There's been several bugs fixed that require you to rebuild the scenario for the fixes to take effect.

In short, rebuild first, then make your changes, then rebuild again. Be sure to save your weapon export file to save the time and trouble of having to manually input ammo loadouts all over again.

27. H3ANW PLATFORM ASSISTANT

The H3ANW Platform Assistant is a collection of spreadsheets in MS Excel format that, allows the user to create platforms and calculate sensor performance for use in H3ANW.This is a **community** supported set of spreadsheets and AGSI is working with the community to update them to v3.10 models and to host the files on the AGSI Wiki. The exact names and functionality of the spreadsheets is subject to change.

No warranty nor support is provided as these are Community created and supported.

Harpoon H3ANW Radar Assistant - A collection of spreadsheets in MS Excel that models air and surface search radar performance along with a series of spread sheets to provide the user with a stealth model, Frequency Range, and Radar Horizon.

H3ANW Sonar Assistant - This collection of Excel sheets will model and simulate sonar performance as used in H3ANW for both active and passive sonar of all frequencies from LF to HF. Grid Layout Sheet - A simple excel sheet to help DB authors set up installations for their database.

Sortie Rate Sheet - A simple sheet to generate sortie rates for aircraft in game.

Strike Planner - Another simple excel sheet that will determine flight and launch times for weapons.

Many of these sheets are currently being worked over and they will be released as time permits.

28. PREFERENCES

28.1. COMMAND-LINE OPTIONS

"A ship-of-war is the best ambassador."- Oliver Cromwell

You can use command-line options in H3ANW to activate several additional options (Skipping the Intro, Auto Save, Show Weapons-calculations and error-logging). The different command line options are as follows: Normally handled in the H3ANW Launcher

-? Displays a list of valid command-line switches.

-a or -A Auto Save. This will autosave the game every couple of minutes. The savegame will be called Auto.sav and is located in your H3ANW directory. (3.7.0) The autosave file name is the name of the session ("Session" in SP) postpended with "-AUTO-SAV".

-I<path> Allows you to specify an alternate path to the Harpoon3.ini file (e.g., C:\Harpoon3\myIniDir)

-lx or -Lx Allows you to specify the maximum length of a simulation cycle in seconds. x must be an integral divisor of 15 (1, 3, 5, or 15), default is 15 (example: -L15). Formerly, the simulation would run in steps of the lesser of the selected time compression, or 15 seconds (the length of a paper-rules air phase). This switch allows you to set a lower ceiling on the size of the time step, making the simulation more precise but possibly slower.

-s<section> Allows you do select an alternate section of the Harpoon3.ini file to override

or settings in the default section.

-S<section>

-t or -T Skip Intro. This will load H3ANW without displaying the Intro-animation

-v<mode> Override the Harpoon3.ini video setting. <mode> is the VESA mode corresponding to the resolution you wish to use. See table below for valid settings. 3.7.1 - <mode> is more flexible, see notes accompanying VESA table.

-V Display the version information for the executable.

-w or -W Show Weapons Calculations. This will display the weapons calculations in the message window.

28.1.1. USING MULTIPLE COMMAND LINE ARGUMENTS

You can combisne the different command line options by simply adding them in the end. Harpoon3.exe -t -a for example will skip the intro and make autosaves. You might want to create a .bat file with the command line options so you don't have to open a DOS-window to start H3ANW.

28.1.2. VESA MODE DEFINITIONS FOR -V

VESA Mode Resolution

101	640 x 480
103	800 x 600
105	1024 x 768
107	1280 x 1024

28.1.3. OTHER LEGAL VALUES FOR -V

You may also use values for VideoMode and VideoWidth as documented in the .ini file. The program is able to determine which type of value you are using. Both will set the width of the program display using <mode> (if it is 640 or higher, it will be used directly as a pixel count, if it is at least 4 but less than the VESA codes, it will be multiplied by 160 to provide the pixel count), and the height of the display will be 3/4 the width (4:3 is the standard aspect ratio for monitors). If you want an aspect ratio of other than 4:3, you must use the Harpoon3.ini file settings for VideoWidth and VideoHeight.

28.2. OPT FILES

"No man will be a sailor who has contrivance enough to get himself into a jail; for being in a ship is being in a jail, with the chance of being drowned... a man in a jail has more room, better food, and commonly better company."

- Samuel Johnson

The second mechanism of change within the game is the option files (.opt). AGSI has included these files with the current build of the simulation to allow you to turn certain features on and off. If you look within your H3ANW directory you will see a folder named options (\options). Within that folder you will see a folder named disabled (\options\disabled) and in each you will find the files that correspond to each available option which I will describe in detail below. To activate options you simply leave the corresponding files in the option folder. To deactivate options you move the corresponding file to the disabled folder. Normally handled in the H3ANW Launcher

AALog.opt - This file enables usage of the After-Action Log, which will be written in the "AA logs" folder (\AALogs) when activated.

ExtrashortPointDefense.opt – This option provides a little extra information about point defense, but keeps the messages short to just tip off the player that certain things are being used (e.g. you will see the worlds "buzz" or "zap" when various types of ECM are being used).

ExtraVerbosePointDefense.opt – Allows every computation involved in point defense to be included in the AA log.

Moviemaker.opt – This file enables the auto-screen grab feature. Edit the MovieTime.txt-file to change the time between auto-screen grabs.

RuninWindow.opt – This file allows H3ANW to be run in a window rather than full screen mode. Removed in version 3.7.1, use the .Launcher instead.

ShowPointDefense.opt – This prints out some basic information to the AAlog and must be active for ExtraShortPointDefense.opt and ExtraVerbosePointDefense.opt to work.

UseNukes.opt – This file grants nuclear release. This option should only be used in scenarios designed for nuclear exchange.

VerboseWeaponDetection.opt – If you have this option enabled you will receive extra feedback about weapon detections and targeting. You must have the AA Log option enabled or be in watch mode.

29. INDEX

30. CREDITS

V3.10 SPECIFIC CREDITS

ADVANCED GAMING SYSTEMS

LEAD PROGRAMMER/PRODUCER
Russell "Rusty" Sharp

PROGRAMMING
Rusty Sharp, Darrel Dearing

QUALITY ASSURANCE
Kit Daniels, Nithin Sundararajan, Dale Hillier, Darren Buckley, Mike Price, REPSIM, and Adam Nagao

BETA TESTING
Christopher "Kip" Allen, Darren Buckley, Dale Hillier, Harold Hutchison, Ralf Koelbach, Jeff Krump, Jan Masterson, Steve Mills, Craig Paffhausen, Steve Reynolds, Freek Schepers, Illias Sellountos, Gregg Smith

DATABASES
Dale Hillier (ANWDB), Darren Buckley (HUD3)

MANUAL UPDATES
Much of the Advanced manual was created from the original H2 help files (created by Three Sixty Pacific) and the AGSI Wiki, extended by documentation created/organized by Dale Hillier. David Young helped re-organize all of the manuals. Don Gilman did most of the editing with lots of help from: Rusty Sharp, Dale Hillier, and Darren Buckley.

INTRO MUSIC
Rachel Sturtevant

SUPPORTING ART, WEBSITE REWORK
Updated Command Card, new product badges and lots of small art pieces whenever we asked Megan "Dragon Flame" Bednarz

ORIGINAL PLATFORM ARTWORK
Philip Alexander

EXECUTIVE PRODUCER
Don "Uncle Harpoon" Gilman

SYSOP
Sean Whitehead

OTHER CONTRIBUTING PLAYERS:
Kristina Whitcomb, David Whitehead, Christopher Irvine, Harold Hutchison, and Ralf Koelbach.

WITH ONGOING SUPPORT
Larry Bond and Chris Carlson the creators of the Harpoon and Admiralty Trilogy Systems – the heart of our product.

MATRIX GAMES

EXECUTIVE PRODUCER
David Heath

ASSOCIATE PRODUCER
Erik Rutins

ART DIRECTOR
Marc von Martial

BOX AND LOGO DESIGN
Marc von Martial

MANUAL EDITING AND CONTENT
Sean Drummy

MANUAL DESIGN AND LAYOUT
Marc von Martial

PUBLIC RELATIONS & MARKETING
Sean Drummy

PRODUCTION ASSISTANT
Andrew Loveridge

DIRECTOR OF OPERATIONS AND ACQUISITIONS
David Heath

Larry Bond's HARPOON ULTIMATE EDITION

LEAD ADMINISTRATION
Liz Stoltz

MANAGER OF DEALER RELATIONS & BUSINESS DEVELOPMENT
Karlis Rutins,

CUSTOMER SUPPORT STAFF
Daniel Heath, Alex Fiedler

FORUM ADMINISTRATION
Erik Rutins, Alex Fiedler, Marc von Martial, David Heath

WEB-DATABASE DESIGN & DEVELOPMENT
Alex Fiedler

NETWORK AND SYSTEM ADMINISTRATOR
Alex Fiedler, David Heath

SOCIAL MEDIA MANAGER
Steve Stafford

PC SUPPORT AND NETWORK ADMINISTRATOR
Lance Stoltz

QUALITY ASSURANCE LEAD
Erik Rutins

SPECIAL THANKS
To our new colleagues at Slitherine Ltd.: JD McNeil, Iain McNeil, Marco Minoli, Philip Veale, Andrea Nicola, Richard Evans, Christian Bassani.

MATRIX NEXGEN
Alexander Rutins, Andrew Heath, Nicholas Heath, Shane Heath, Austin Stoltz, Noah Stoltz, Jesse Stoltz, Heidi Fiedler, Blake Fiedler, Harold Dupree

OUR STRENGTH
We thank God for giving us the ability and strength to complete this project and follow our dream.

We would also like to thank our families and friends for giving us their non-stop love and support during this project.

Slitherine Ltd and Matrix Games Ltd

END USER LICENSE AGREEMENT

READ THIS SOFTWARE LICENSE AGREEMENT ("LICENSE") CAREFULLY BEFORE PROCEEDING TO INSTALL THE SOFTWARE. BY PRESSING "AGREE," YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, PRESS "DISAGREE" THIS LICENSE AGREEMENT IS A LEGALLY BINDING CONTRACT BETWEEN YOU AND SLITHERINE LTD. AND/OR ITS SUBSIDIARIES, AFFILIATES OR SUB LICENSEES.

1. **General.** This software product is copyrighted and is protected by international law. The software and any accompanying documentation or media including this License whether on disk, in read only memory, or in any other form are licensed, not sold, to you by either Slitherine Ltd or Matrix Games Ltd and is for use only under the terms of this License. Slitherine and or Matrix Games reserves all rights not expressly granted to you. The rights granted herein are limited and do not include any patents or intellectual property rights. Slitherine and or Matrix Games expressly retain ownership of the Software itself.
2. **Permitted License Uses and Restrictions.** This License allows you to install and use one copy of the Software on a single computer at a time. This License does not allow the Software to exist on more than one computer at a time, and you may not make the Software available over a network where it could be used by multiple computers at the same time. You may not copy, reproduce, translate, decompile, reverse engineer, disassemble, modify, or create derivative works from the assembled code or any part thereof. The software may contain an Editor that allows purchaser to create new assets, modify existing assets or files or create custom levels, scenarios or other materials for use solely in connection with the existing software ("new materials"). Purchaser is not permitted to use, or allow third parties to use the Editor and/or any new materials created, for any commercial purposes whatsoever, or in any other software, without the express prior written permission of Slitherine and or Matrix Games.
3. **Game Servers.** Use of the software by purchaser on Slitherine or Matrix Games servers is allowed entirely at the discretion of Slitherine and or Matrix games, who at their sole discretion reserve the right to remove, deny or prevent the purchaser from using the Companies servers for any reason whatsoever including unreasonable, abusive or offensive language or behaviour and without consultation or notice.
4. **Transfer.** Purchaser may not rent, lease, lend or sublicense the Software.
5. **Termination.** This License is effective until terminated. Your rights under this License will terminate automatically without notice from Slitherine or Matrix Games if you fail to comply with any term(s) of this License. Upon the termination of this License, you shall cease all use of the Software.
6. **Warranty.** This Software is provided without warranty of any kind, whether express or implied, including warranties of merchantability and fitness for a particular purpose, which are hereby disclaimed. In no event will Slitherine or Matrix Games be liable for any special, incidental, or consequential damages resulting from possession, use, or malfunction of this software product.
7. **Disclaimer.** You expressly acknowledge and agree that use of the software is at your sole risk and that the entire risk as to satisfactory quality, performance, accuracy and effort rests with you. The software is provided "as is"; with all faults and without warranty of any kind, and Slitherine, Matrix Games and their affiliates or licensors hereby disclaim all warranties and conditions with respect to the software, express, implied or statutory. Slitherine and or Matrix Games do not warrant against interference of your enjoyment of the software, nor that the functions contained in the software will meet any requirements, nor that the operation of the software will be uninterrupted or error-free, or that defects in the software will be corrected. No oral or written information or advice given by Slitherine and or Matrix Games or any authorized representative shall create a warranty.

Should the software prove defective, you assume the entire cost of all necessary servicing, repair or correction.

8. **Limitation of Liability.** Is restricted to the full extent not prohibited by law, in no event will Slitherine and or Matrix Games be liable for personal injury, or any incidental, special, indirect or consequential damages whatsoever, including, without limitation, damages for loss of profits, loss of data, business interruption or any other commercial damages or losses, arising out of or related to your use or inability to use the software, however caused, regardless of the theory of liability (contract, tort or otherwise) and even if Slitherine or Matrix Games has been advised of the possibility of such damages. In no event shall Slitherine and or Matrix Games total liability to you for all damages (other than as may be required by applicable law in cases involving personal injury) exceed the amount which the purchaser paid for the software or Fifty US Dollars ($50) whichever is less. The foregoing limitations will apply even if the above stated remedy fails in its essential purpose.
9. **Controlling Law and Severability.** This License will be governed by and construed in accordance with the laws of England and Wales. If for any reason a court of competent jurisdiction finds any provision, or portion thereof, to be unenforceable, the remainder of this License shall continue in full force and effect.
10. **Complete Agreement; Governing Language.** This License constitutes the entire agreement between the parties with respect to the use of the Software licensed hereunder and supersedes all prior or contemporaneous understandings regarding such subject matter. No amendment to or modification of this License will be binding unless in writing and signed by Slitherine and or Matrix Games. Any translation of this License is done for local requirements and in the event of a dispute between the English and any non-English versions; the English version of this License shall govern.