Development of an Early Stage Ship Design Tool for Rapid Modeling in Paramarine

by

Eric J. Thurkins Jr.

Bachelor of Science in Physics
University of Massachusetts-Lowell, 1999

Master of Education
University of Massachusetts-Lowell, 2002

Master of Business Administration
The Citadel, 2008

Submitted to the Department of Mechanical Engineering
In Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer

and

Master of Science in Mechanical Engineering

at the
Massachusetts Institute of Technology
June 2012

Signature of Author [illegible]
Eric J. Thurkins Jr.
Department of Mechanical Engineering
May 8, 2012

Certified by [illegible]
Chryssostomos Chryssostomidis
Professor of Mechanical and Ocean Engineering
Thesis Supervisor

Accepted by [illegible]
David Hardt
Chairman, Committee for Graduate Students
Department of Mechanical Engineering

(THIS PAGE INTENTIONALLY LEFT BLANK)

Development of an Early Stage Ship Design Tool for Rapid Modeling in Paramarine

Eric J. Thurkins Jr.

**Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degrees of**

Naval Engineer
and
Master of Science in Mechanical Engineering

Abstract

In early-stage ship design, it is helpful to perform preliminary design and analysis on many configurations to assist in developing and narrowing the trade space. This process is further complicated with the increasing interest in concepts that are breaks from previous practice, such as Integrated Power System (IPS) designs, which require initial development to go deeper than historically based parametrics can provide. Paramarine is a ship design and analysis tool which can be used in this early-stage design; however, as with many early-stage design tools, the fleshing out of diverse ideas in Paramarine can be time and resource consuming. In an effort to enable a developer to create early-stage designs with depth significant enough to be meaningful but still general enough to allow the level of flexibility in design required in the early stages of development, this project seeks to develop an Early Stage Ship Design Tool (ESSDT). This ESSDT is a novel interface with which a designer can rapidly develop and alter basic, major design components of a ship from a compiled database of components and gain a rendered model for analysis within the naval design tool Paramarine. By making use of many early-stage parametric and developed calculations and leveraging the use of IPS, this ESSDT automates many of the initial ship's estimates and minutia of design. Utilizing both Excel and Paramarine software, the ESSDT rapidly creates a visual model of a basic naval vessel with primary systems and equipment from realatively few initial user inputs while embodying a depth of user-changeable default settings for more complex and non-standard design efforts.

Several case studies were run to show the capability and flexibility of the tool, as well as showing how new powering and mechnical systems can affect the parameters of the ship as a system of systems.

Thesis Supervisor: Chryssostomos Chryssostomidis
Title: Professor of Mechanical and Ocean Engineering

(THIS PAGE INTENTIONALLY LEFT BLANK)

Table of Contents

List of Figures

List of Tables

Acronym List

AMR	Auxiliary Machinery Room
ASSET	Advanced Surface Ship Evaluation Tool
CAD	Computer-Aided Design
CASD	Computer-Aided Ship Design
CO	Commanding Officer
CSDT	Cooling System Design Tool
ESRDC	Electric Ship Research and Design Consortium
EST	Equipment Selection Table
FP	Forward Perpendicular
IPS	Integrated Power System
IPSDM	Integrated Power System Design Module
LBP	Length Between Perpendiculars
MCT	Module Construction Table
MIT	Massachusetts Institute of Technology
MMR	Main Machinery Room
NAVSEA	Naval Sea Systems Command
OMR	Other Machinery Room
POSSE	Program Of Ship Salvage Engineering
QHM	Quick Hull Method
RCS	Radar Cross-Section
SFC	Specific Fuel Consumption
SWBS	Ships Work Breakdown Structure
VBA	Visual Basic for Applications
VLS	Vertical Launch System
XO	Executive Officer

Biographical Note

Eric Thurkins is an active duty Navy Lieutenant currently pursuing degrees in Naval Engineering and Mechanical Engineering as prerequisites for subsequent assignment as an Engineering Duty Officer.

LT Eric Thurkins graduated from the University of Massachusetts at Lowell in 1999 with a Bachelor's degree in physics and in 2002 with a Master's degree in education. After five years as a high school physics teacher at Nashua Christian Academy in Nashua, NH, he received his Navy commission in October of 2004 and attended Officer Indoctrination School in Newport, RI.

In December of 2004, he reported aboard the Naval Nuclear Power Training Command (NNPTC) in Goose Creek, SC as an instructor in the Enlisted Physics division. He then went on to teach in the Officer Math and Physics division, and eventually served as that division's director. While at NNPTC, he also obtained his MBA from The Citadel in Charleston, SC.

In 2008, LT Thurkins was selected for lateral transfer to the Engineering Duty Officer community and for admittance to the Massachusetts Institute of Technology (MIT). He reported to MIT's 2N program for naval engineering in May of 2009.

LT Thurkins' awards include a Navy Commendation Medal, Navy Achievement Medal, and recognition as a Master Training Specialist.

Acknowledgements

I would first and foremost like to thank Megan Thurkins, my loving wife of twelve years. Were it not for her acceptance of me taking on yet another degree and all the ensuing familial travails, this thesis would not have been possible. From her willingness to let "mundane matters" such as the upkeep of our house go by the way-side these past several months, to her readiness to read through page upon page of technical discourse to root out my numerous spelling and grammatical foibles, she has been a greater help to me than I could ever express.

I would also like to thank Professor Chryssostomos Chryssostomidis and Julie Chalfant. Their guidance and insight were essential in equipping me to complete this task. Without their helpful direction and wisdom, this thesis would have never even seen the light of day.

It would also be remiss of me to not mention, with the greatest gratitude, the collaboration and effort on the part of David Jurkiewicz in his co-development of the Integrated Power System Design Module. The give and take of our discussions and his continuous presence as a sounding board for ideas provided an invaluable part of the creative design process. Without him this thesis would have, at best, been a mere shadow of what it is.

And finally, I wish to thank my daughters, Lyndley, Rosalie, and Grace, who had to put up with so many evenings of, "Sorry, I can't play Lego Harry Potter, I have to work," and other such similar statements. I hope to give them many hours of my evenings from here on out.

Development of an Early Stage Ship Design Tool for Rapid Modeling in Paramarine

Eric J. Thurkins Jr.

Submitted to the Department of Mechanical Engineering
In Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer
and
Master of Science in Mechanical Engineering

Executive Summary

In the early stages of ship design, it is often desirable to rapidly model various configurations of vessels in an attempt to narrow the trade space for more in-depth analysis and design. There are a number of tools available to the modern naval architect which accomplish this goal. These include the Advanced Surface Ship Evaluation Tool (ASSET), the Max Surf suite of programs, Program of Ship Salvage Engineering (POSSE), and Graphic Research Corporation's (GRP) Paramarine program among many others. However, many of these programs rely heavily on parametrics based on traditionally structured ships with conventional engine room layouts involving direct mechanical linkages to main engines via a reduction gear system. As world navies begin moving to Integrated Power Systems (IPS), these tools' parametrics will need to be re-evaluated. In addition, as ship design is a very complex pursuit, many of these tools involve a lengthy series of input parameters to construct even a relatively simple ship design from the ground up.

The ESSDT attempts to address both of these issues. The ESSDT is a Visual Basic for Applications (VBA) based program which interfaces with both Excel and Paramarine software to generate and analyze an initial stage early ship design. The data entry portion of the program is accomplished in Excel. The user manually inputs several key ship's parameters, selects a pre-constructed engine room arrangement, selects large-scale equipment and its placement, and uses basic geometry to synthesize a deckhouse structure. The execution of the tool then generates a visual and characteristic model within Paramarine for more in-depth analysis.

To accomplish this process, the ESSDT makes use of both a concurrently-developed tool for construction of shipboard machinery spaces and an inclusive equipment library consisting of large scale shipboard components available for inclusion in the design. The tool for developing machinery spaces enables the user to construct the Main Machinery Rooms (MMRs), Auxiliary Machinery Rooms (AMRs), and Other Machinery Rooms (OMRs) for a naval vessel's powering requirements. The library is designed to include extensive information about large scale ship-board components including basic clearance dimensions, weight, power, cooling, and graphics.

The ESSDT also makes use of a banking system to account for volume, weight, and power. Based upon initially input parameters, it determines what a particular design has available for weight, power, and volume within the ship. It then graphically displays this information along with what the current design requires for each of these parameters. This allows for rapid design decision making in the earliest of design stages.

To show the reliability and capability of the ESSDT, two case studies were run. The first case study involved inputting the large scale design parameters of an *Arleigh Burke* flight IIA class destroyer and comparing the ESSDT's output to that of an actual *Arleigh Burke* flight IIA class destroyer. This test measured the ability of the ESSDT to be an effective predictor of eventual ship design parameters at an early stage of design development. It was found that the ESSDT was able to produce a model in Paramarine with dimensions and sizing within 5% of an *Arleigh Burke* flight IIA class destroyer with all major equipment in place in under two hours time.

The second case study involved starting with developing a medium-sized surface combatant of around 9500 LT and then changing its engine room configuration to a physically smaller configuration with lower power output and comparing the capability of the two resulting ships. This test demonstrated the ability of the ESSDT to be used as a tool in the early stages of ship design to make trade-space trade-offs very early on. It was found that the two ships ended up fairly comparable in both size and capability. However, with an expansion of the model hulls available for the ESSDT to draw from, it is believed that more significant results could be rapidly developed.

In conclusion, the ESSDT showed its merit as a tool for naval architects and engineers for early stage design and analysis of potential ship designs.

(THIS PAGE INTENTIONALLY LEFT BLANK)

1.0 Introduction

This research applies the principles of Naval Architecture and Mechanical Engineering to the early stage design of surface warships and their ship-board systems. The primary research objective was to build an Early Stage Ship Design Tool (ESSDT) for Naval Architects that can rapidly generate system visualizations and model characteristics of warships.

1.1 Organization of Thesis

This thesis contains five chapters (Introduction, Design Tool Use, Design Tool Concepts, Case Studies and Tradeoffs, and Conclusions) and five appendices. The Introduction provides background on the hypothesis and motivations behind the topic. The Design Tool Use chapter is an upper level discussion of the tool providing an overview of the tool, the structure and use of its inputs and libraries, the output product into Paramarine, and potential applications. A greater look at the minutia of the tool is in the Design Tool Concepts chapter which details the structure and calculations behind the Excel interface; a bank system for tracking required and available volume, weight, and power; the synthesis of hulls within Paramarine; and the Paramarine system structure. The Case Studies and Tradeoffs chapter runs through several comparison scenarios modeled by the ESSDT. The final chapter, Conclusions, highlights the results of the case studies, utility of the tool, and points out areas for future research. The attached appendices contain the ESSDT instruction manual, sample images of the input and output pages, the resultant Paramarine file structure, input for the case study, and the VBA code for the tool.

1.2 Topic Motivation

There are two current ways in which the present field of early stage ship design tools are lacking. The first is the complexity and depth of the initial synthesis of designs, requiring a fairly substantial background in naval architecture and substantial input of ships' system data. The second issue is with regards to the rise of Integrated Propulsion Systems (IPS) in which the main propulsion unit is not mechanically coupled to the ship's main engines but is electrically connected and driven. Many of the simpler early stage design programs do not allow the flexibility in design which these relatively new IPS systems afford.

There are currently many programs on the market for the early stage design and analysis of warships. Among the more versatile and powerful of these tools is Graphic Research Corporation's (GRC) program Paramarine. However, as with many of the more sophisticated options for these tools, constructing a ship within them and pulling in all pertinent equipment data can be a long, laborious task. The beginning tutorial (Pawling, 2009) can easily take over 40 hours to complete, and the result is a fairly standard vessel composed, primarily, of components which happen to already be present in the tutorial's preliminary file structure. To design a complete ship from the ground up with all substantial components could easily be a far lengthier process.

On the other end of the spectrum are simpler programs like the Advanced Surface Ship Evaluation Tool (ASSET). This tool allows for a relatively rapid synthesizing of early stage design. However, it is greatly based on historic parametrics for traditional propulsion vessels and does not allow a great deal of flexibility for non-traditional designs such as the incorporation of IPS. In addition, in order to thoroughly generate and evaluate the efficiency of the designs generated in ASSET, a good deal of fairly expert knowledge and experience within the program is required.

Based on the complexity on one end of the spectrum of early stage design tools and the lack of flexibility on the other, it was determined that there was a potential need for a program with a ship-board equipment library which could rapidly synthesize early stage ship plans within a more complex tool such as Paramarine while incorporating a simpler interface which allows more rapid and intuitive placement of ship's components.

1.3 Background

The development of computer-aided design of ships has revolutionized the naval architects' approach to ship design over the past half century. As design has moved from the draft table to the keyboard, computerized models have become more complex and accurate. This has developed the ability to analyze more and more initial designs to greater depth without ever having them leave the drawing board. However, in an attempt to further, quickly examine how perturbations in design can affect examples within a design space, a tool allowing rapid prototyping of these early stage designs is needed to assist ship and systems engineers to plan for future designs.

1.3.1 The History Behind Computer-Aided Ship Design Tools

In 1952, a computer-aided ship design group was first established by the Naval Sea System Command (NAVSEA), only one year after NAVSEA received their first computer (Carlson, 1982). One of the first recorded uses of the phrase, "Computer-Aided Ship Design," (CASD) was at a U.S. Navy Symposium in Washington D.C. in 1966 (D.W.Taylor Model Basin, 1966). Pre-dating this had been developments in various areas of Computer-Aided Design (CAD) technologies. The tri-fold developments of the digital media for manufacturing automata's numerical control, the ability to represent a ship's geometry digitally, and the use of computers for the computationally complex tasks of ship design calculations (Adams, 2012) were necessary precursors to CASD.

By the mid 60's, all needed items were in place to pave the way for CASD. NAVSEA first developed the Computer-Aided Ship Design and Construction (CASDAC) program in 1966. Its purpose was to, "prove the feasibility of computer application, to verify the benefits, and to foster the use of computers to all phases of ship design and construction." (Carlson, 1982) This led to programs such as BRITSHIPS (Hurst, 1973) and FORAN (Belda, 1973) in the early 70's. Then, commissioned in 1977 (Wikipedia, 2012), the *USS Oliver Hazard Perry* was the first warship designed by computer (Adams, 2012).

Since that time, CASD has continued to evolve. Additionally, it has influenced the direction of ship design, methodology, and analysis. By freeing the naval architect from much of the minutia of design and allowing rapid determination of ship's specification, CASD allows an order of magnitude increase in the number of early stage designs which can be explored. CASD has allowed a dramatic expansion in the ability of the naval architect to peruse better ship optimization in the midst of the inherent conflict of design constraints and optimization criteria (Papanikolaou, 2009).

However, a trained naval engineer is still required to make many design decisions in the process of using CASD. There have been some attempts to automate some of the design processes. Work at the University of Michigan has made progress in developing a tool utilizing fuzzy logic to optimize arrangements of space within a ship (Nick & Parsons, 2007). Work at MIT has led to a tool developed to plan the cooling system for a naval combatant (Fiedel, 2011). Both of these tools, though, require an initially designed ship to implement their framework. There is, currently, no tool to allow a user to rapidly create a visual model of a basic naval vessel with primary systems and equipment from realatively few initial user inputs while embodying a depth of user-changeable default settings for more complex and non-standard design efforts.

In addition to this lack of design capability with the current generation CASDs, many of the ones with fewer inputs and simpler interfaces do not allow for the recient advent of IPS for shipboard propulsion.

In a traditional ship's propulsion system, the preliminary source of motive power, the prime mover, is directly coupled to a set of reduction gears which are, in turn, coupled to a propeller. Electrical power for the ship is provided by a generator which is also shafted to another prime mover to provide propulsive power as can be seen in Figure 1.

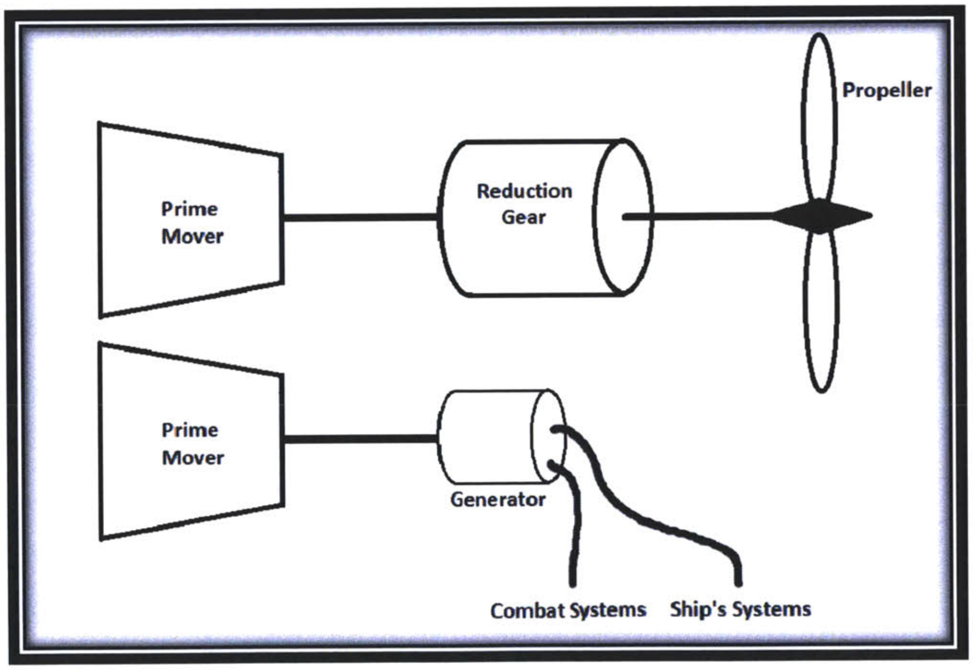


Figure 1: Traditional Ship's Propulsion System

technologies were still not quite up to the challenge of delivering ship propulsive power in a manner better than the traditional architecture (Bowman, 2000).

Figure 3: USS Buckley Steaming Under Electric Power (Wikipedia, 2012)

The recent advances in solid state technology and electric motors have now made electric propulsion systems a viable possibility. The DDG-1000, due to be completed in 2014 (Wikipedia, 2012), will be a truly innovative IPS driven ship. With the resurgance of electrically driven ships will come the need for design tools which can accommodate these flexible, inovative design options in a manner which will allow for rapid development of preliminary designs to analyze a previously under-developed design space.

By contrast, an IPS has its primary mover directly coupled to a generator. This generator, in turn, provides power to the ship's systems and an electric propulsion motor as seen in Figure 2. Although not as energetically efficient as a traditional plant, IPS allows for much greater flexibility of design and equipment placement since a ship's layout is no longer dependant upon the deliniated stacklength requirements of the traditional layout. It also allows for greater flexibility of operations related equipment needs. The power provided by primary movers no longer has to be capable of full power on both propulsion and ship's systems. If a higher load item like a rail gun is to be employed, and it is possible to operate the ship at speeds less than its maximum rated speed, power can be shunted from propulsion to other systems. It is not possible to do this with a traditional propulsion system.

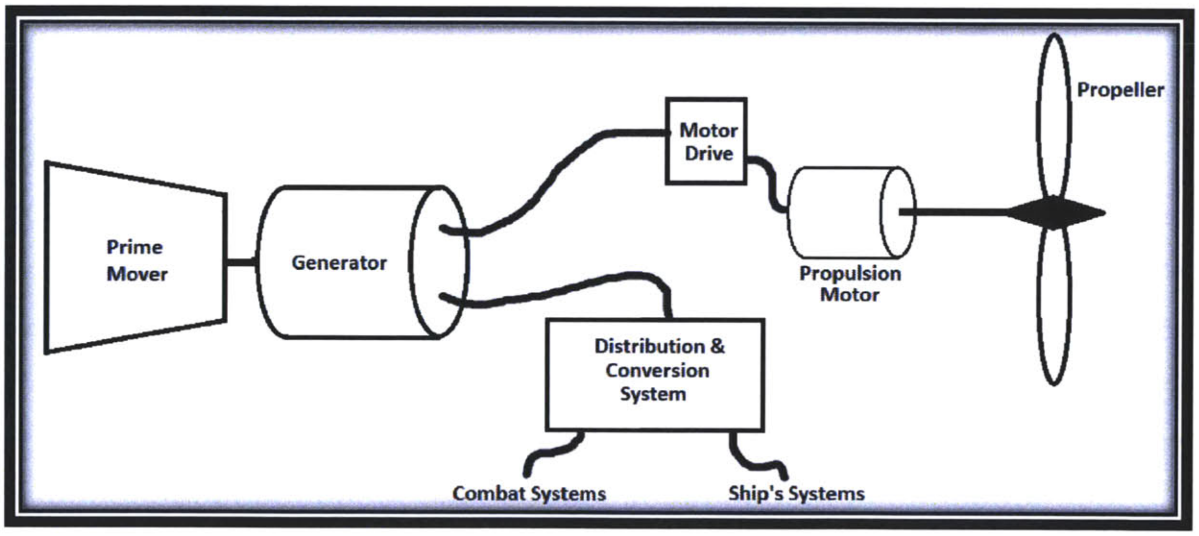


Figure 2: IPS Propulsion

The history of electric propulsion goes back quite a ways. Launched in 1912, the USS Langley was the first naval vessel to utilize an electric propulsion system (MIT Sea Grant College Program, 2012). Early submarines were, effectively, electric ships as well as the WWII era BUCKLEY class destroyer escorts shown in Figure 3. Unfortunately, these early systems often proved to be unreliable and energy inefficient. Even as reciently as the 1960's and 1970's, with the *USS Tullibee* and *USS Glenard P. Lipscomb*, the power conversion and electric motor

1.3.2 Complexity of Naval Architecture

It is an understatement to say that ship design is complex. The number of systems tied together, interwoven, and condensed into the concentrated package required for ship-board operations make ship design a daunting task. Any program seeking to automate this process must account for a diverse myriad of contingencies and interactions. "Ship design is an iterative process, especially in the early stages. . . The reason for iteration is that ship design has so far proven to be too complex to be described by a set of equations, which can be solved directly. Instead, educated guesses are made as to hull size, displacement, etc. to get the process started and then the initial guesses are modified." (Society of Naval Architects and Marine Engineers, 2003) Figure 4 graphically illustrates the complexity of this iterative process.

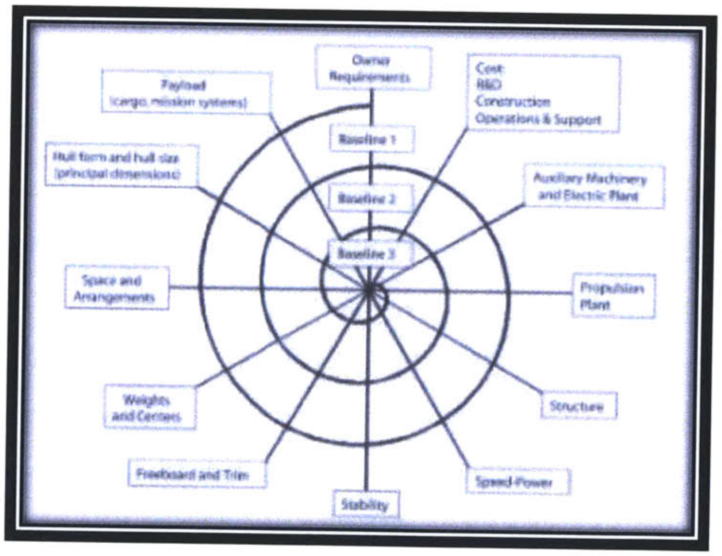


Figure 4: Design Spiral (Society of Naval Architects and Marine Engineers, 2003)

It is daunting, and perhaps even undesirable, to consider the construct of a program to handle the entirety of this process. The very nature of the design spiral used in naval engineering demands constant, interlinked trade-offs and analysis. However, rather than starting with a completely blank slate, it may be possible for a program to at least make a first level "educated guess", a first pass around the spiral if you will, to develop an initial design starting point from a relatively small number of initially input parameters.

1.3.3 Need for a tool

Many of the current CASD programs on the market today require a significant amount of input from a user to construct this "first-pass" ship. But what information is really needed to have an initial picture or understanding of what and early stage design might look like? What are typical customer requirements?

A customer is going to want a ship that will perform a particular task. To that end, the customer may already have in mind what type of large scale equipment the vessel will need, how large a crew it should have, how long it should be able to remain on station, and maybe even a rough idea of how large it should be. To these requirements, a naval engineer much apply his acumen and experience to extrapolate ship size; a possible, initial hull form; power requirements; and super structure sizing.

Once the naval engineer has made a first pass at these upper tier design decisions, the design process can begin in earnest. From there, a naval engineer or architect may begin using a CASD tool to create several first pass designs. Given the amount of information many of these current programs require, this process may involve a significant investment in time and resources to populate this design space. In addition, many of the simpler early stage design tools make heavy use of parametrics based on traditional engine plant layouts. These fail to recognize the flexibility and structure/equipment differences afforded by IPS.

However, going in, customer requirements already set many of those upper tier design decisions. What if a tool could simply take just those upper level design decisions, taking into account the changes IPS has allowed, and quickly produce a first pass early stage design automatically for the naval engineer to start with? This wouldn't remove the necessity for the naval engineer's expertise, but would augment his ability by enabling rapid synthesis of potential design models. This capability could allow for both a larger initial design space analysis leading to a more optimized final design and a more rapid (and, by extension, less expensive) design process.

1.3.4 Present Design Tools Available

One of the primary CASD tools for the US Navy is the Advanced Surface Ship Evaluation Tool (ASSET). ASSET makes use of a number of design modules that it uses to walks the user through to develop a basic ship design. It uses a significant number of parametric models to determine equipment and space requirements to develop a ship model with propulsion machinery layouts, decks, compartments, and deckhouse. ASSET is currently working on employing an IPS module, but as of version 5.3, this has not been fully incorporated. In addition, the number of modules and their requisite inputs far exceeds the limit on input goals the ESSDT is envisioned to encompass. ASSET walks the user through several lengthy modules that require a large number of user inputs to develop and assess a preliminary model.

For later stage design analysis, programs like CATIA© and ShipConstructor© offer the ability to develop complex and intricate ship models with a plethora of analysis options. However, as was the case with the DDG-1000 project, it can take a large staff upwards a year to completely build a ship's model (Fiedel, 2011).

There are several intermediate tools available such as Maxsurf©, Program Of Ship Salvage Engineering (POSSE), and Paramarine. These tools require significant initial design input, starting with effective "blank sheets", but offer fairly flexible and comprehensive analysis of ship design. Paramarine, in particular, is a fairly flexible tool with good initial analysis abilities. However, with no real guidelines or templates when a design is started, initial design within the program can be daunting.

1.4 Thesis Intent

This thesis intends to address the issues discussed in section 1.3.3 in an expedited manner over the tools discussed in section 1.3.4. In order to meet these goals, the developed program should have relatively fewer initial inputs for initial ship synthesis, draw off of a library of equipment and available hull types, be relatively easier to operate, and rapidly generate a ship model of sufficient detail to be meaningful for further analysis, consideration, and alteration. This will allow a user to rapidly investigate the pros and cons of different, large-scale design trade-offs and develop the practicality or impracticality of future ship designs.

(THIS PAGE INTENTIONALLY LEFT BLANK)

2.0 Use of the Design Tool

This chapter highlights a general overview of the process the ESSDT undergoes in its creation of a naval vessel model in Paramarine from the provided equipment library, parametrics, user inputs, and defaults. It then discusses ways in which this type of application to ship design may be employed for future studies and design processes. A more traditional "user's manual" is provided in Appendix A: Program Instruction Manual. Chapter 3 discusses the actual calculations and process which go into ship synthesis in greater detail.

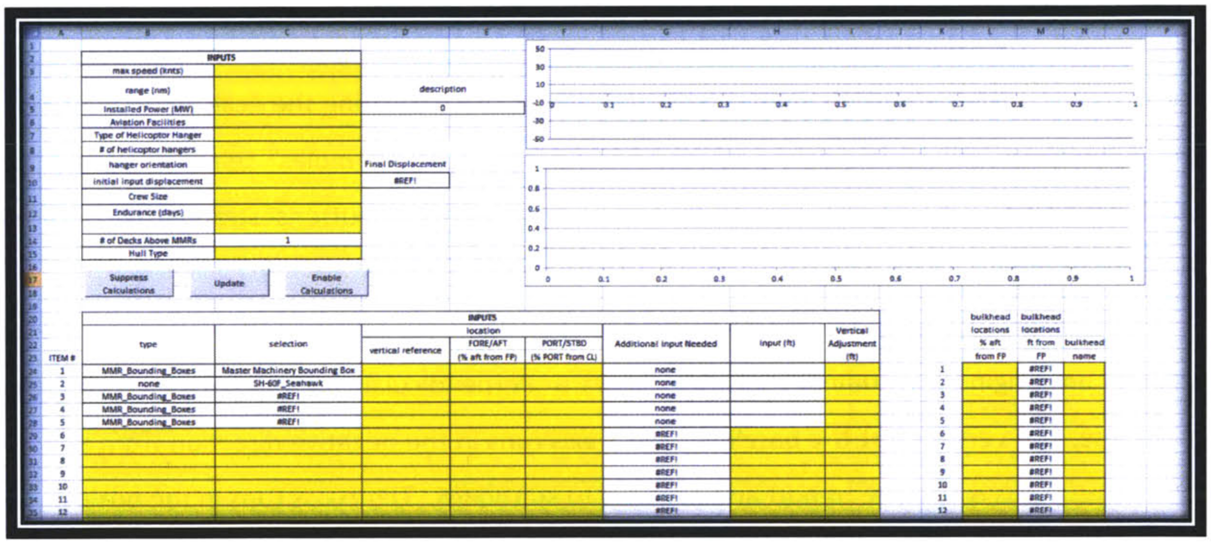
Due to the vast quantity of cross-references and lookup functions within the Excel interface, instantaneous calculations have been suppressed within the program. Although this foregoes onerous wait periods betwixt each data entry, it precludes instantaneous updating based upon whatever user inputs have been changed without, in some way, manually updating the Excel file. This can be accomplished by saving the file, activating the "calculate" option in either the lower left corner of the opened Excel program or "Formulas" selection within the Excel ribbon, or pressing on one of the numerous "Update" buttons strategically located throughout the program.

All ship's locational references within the ESSDT and the constructed Paramarine model assume an origin longitudinally at the Forward Perpendicular (FP), transversely on the centerline, and vertically at the baseline. The x-axis runs in the negative direction from bow to stern. The y-axis is positive to port and negative to starboard. The z-axis runs in the positive-up direction.

2.1 Overview of Process

The key process involves minimizing user inputs necessary for an initial synthesis of a modeled vessel for analysis in Paramarine. There are four key input tabs within the initial Excel user interface. These include "Initial_Input", "Create_Deckhouse", "Adjust_Stacks", and "Import_to_Paramarine". Yellow highlighted regions within each of these sheets represent the allowable user input ranges. There is also a "Defined_Defaults" tab which provides design defaults for model synthesis. These defaults are more in-depth parameters made available for more advanced users with a greater understanding of ship design.

An example of the first of these tabs, "Initial_Input", can be found in Figure 5. In the upper-left hand corner, the user can input the over-arching ship's parameters. These include the items shown in Figure 6. In the bottom portion of this sheet, the user has the ability to select up to 100 major equipment items from the program's library to include in the model as shown in the lower portion of Figure 5. The upper right portion of the screen displays a graph which pictorially constructs the vessel as design decisions are made. Figure 7 shows an example of this figure as the design process progresses. The lower right corner of Figure 5 shows the inputs for bulkhead locations.



INPUTS	
max speed (knts)	
range (nm)	
Installed Power (MW)	
Aviation Facilities	
Type of Helicoptor Hanger	
# of helicoptor hangers	
hanger orientation	
initial input displacement	
Crew Size	
Endurance (days)	
# of Decks Above MMRs	1
Hull Type	

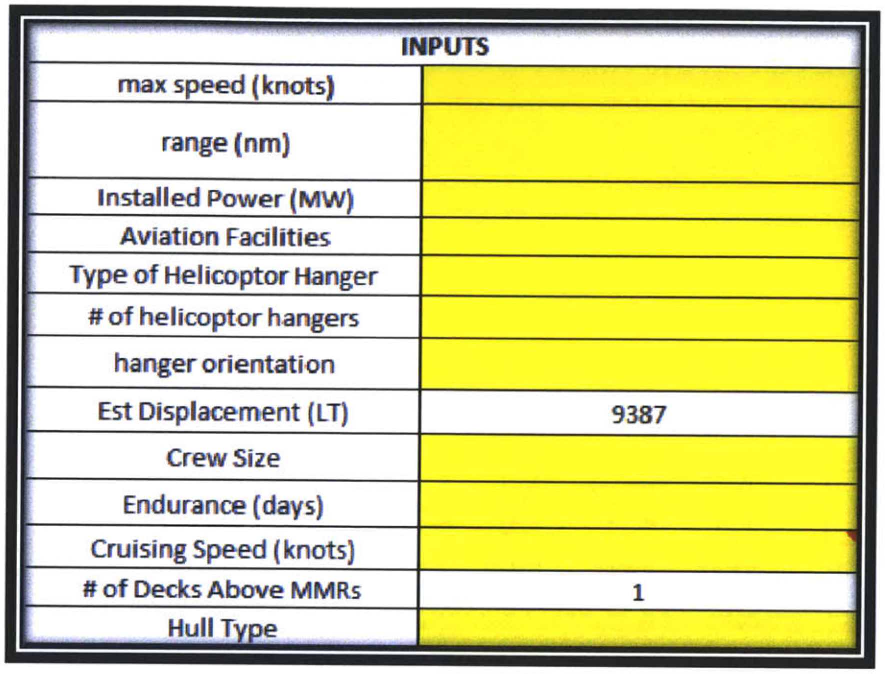
description
0

Final Displacement
#REF!

ITEM #	INPUTS							
	type	selection	location			Additional Input Needed	Input (ft)	Vertical Adjustment (ft)
			vertical reference	FORE/AFT (% aft from FP)	PORT/STBD (% PORT from CL)			
1	MMR_Bounding_Boxes	Master Machinery Bounding Box				none		
2	none	SH-60F_Seahawk				none		
3	MMR_Bounding_Boxes	#REF!				none		
4	MMR_Bounding_Boxes	#REF!				none		
5	MMR_Bounding_Boxes	#REF!				none		
6						#REF!		
7						#REF!		
8						#REF!		
9						#REF!		
10						#REF!		
11						#REF!		
12						#REF!		

	bulkhead locations % aft from FP	bulkhead locations ft from FP	bulkhead name
1		#REF!	
2		#REF!	
3		#REF!	
4		#REF!	
5		#REF!	
6		#REF!	
7		#REF!	
8		#REF!	
9		#REF!	
10		#REF!	
11		#REF!	
12		#REF!	

Figure 5: Initial Input Sheet



INPUTS	
max speed (knots)	
range (nm)	
Installed Power (MW)	
Aviation Facilities	
Type of Helicopter Hanger	
# of helicopter hangers	
hanger orientation	
Est Displacement (LT)	9387
Crew Size	
Endurance (days)	
Cruising Speed (knots)	
# of Decks Above MMRs	1
Hull Type	

Figure 6: Table of initial User Inputs

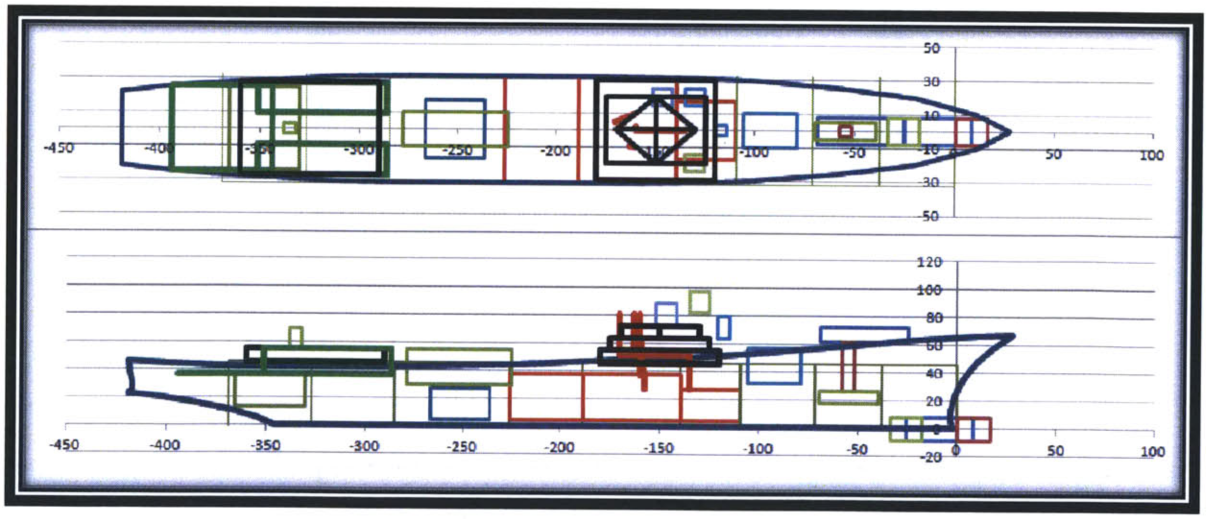
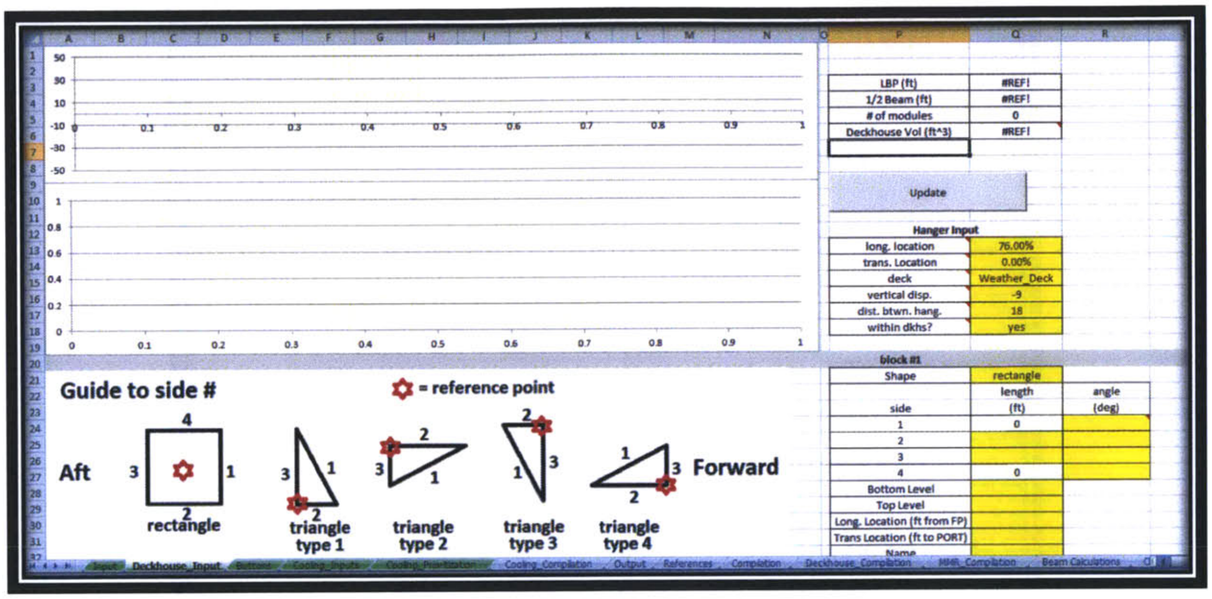


Figure 7: Pectoral Representation of the Design in Process

Figure 8 illustrates the input for the deckhouse construction in the "Deckhouse_Input" tab. The initial input, as seen in Figure 9, has the user specified parameters for the aviation facilities if they were chosen to be present in the initial selection on the "Initial_Input" tab.



LBP (ft)	#REF!
1/2 Beam (ft)	#REF!
# of modules	0
Deckhouse Vol (ft^3)	#REF!

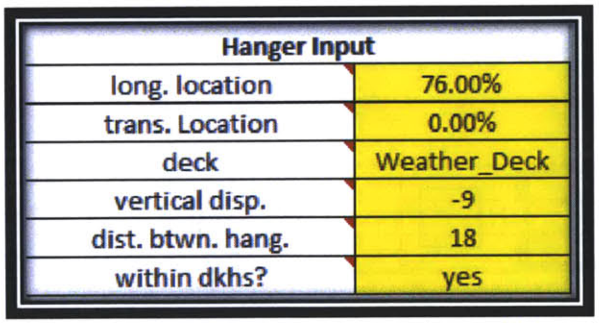
Hanger Input

long. location	76.00%
trans. Location	0.00%
deck	Weather_Deck
vertical disp.	-9
dist. btwn. hang.	18
within dkhs?	yes

block #1

Shape	rectangle	
side	length (ft)	angle (deg)
1	0	
2		
3		
4	0	
Bottom Level		
Top Level		
Long. Location (ft from FP)		
Trans Location (ft to PORT)		
Name		

Figure 8: Deckhouse Input Sheet



Hanger Input	
long. location	76.00%
trans. Location	0.00%
deck	Weather_Deck
vertical disp.	-9
dist. btwn. hang.	18
within dkhs?	yes

Figure 9: Hanger Input

The deckhouse structure itself is constructed form a combination of rectangular and triangular structures as selected by the user. Figure 10 demonstrates these different shapes and how their dimensions are referenced by the interface shown in the lower right corner of Figure 8. A more detailed view of this input can be seen in Figure 11. The option is given of selecting an angle for each of the sides to assist with reduced Radar Cross-Section (RCS). At this time, this is not incorporated into the main ESSDT in a way that allows these deckhouse angles to translate into the Paramarine model. However, it is hoped that, in the future, this function can be incorporated into later iterations of this program.

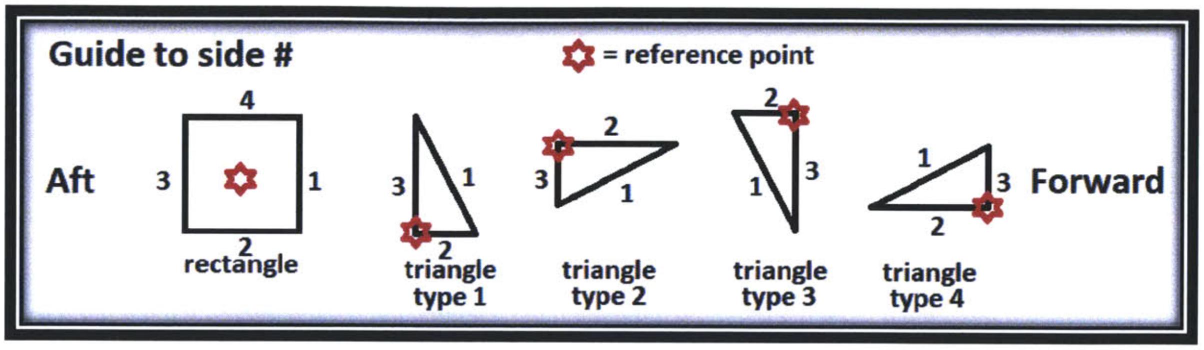
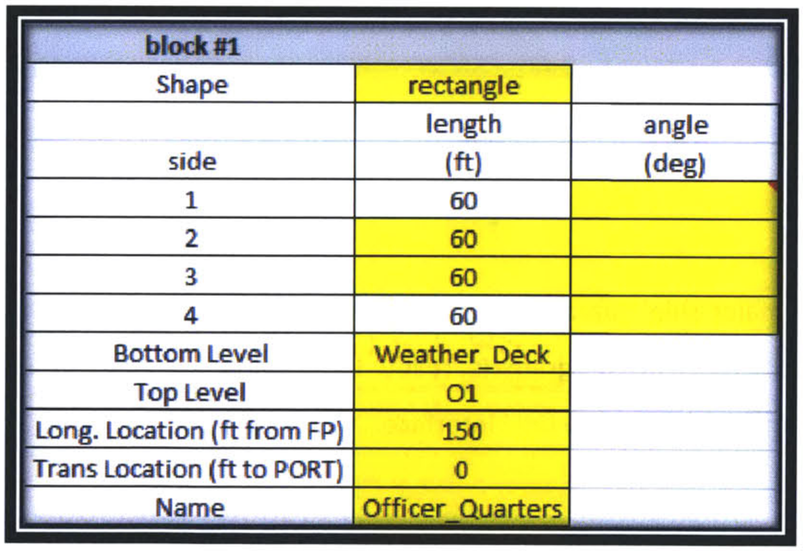


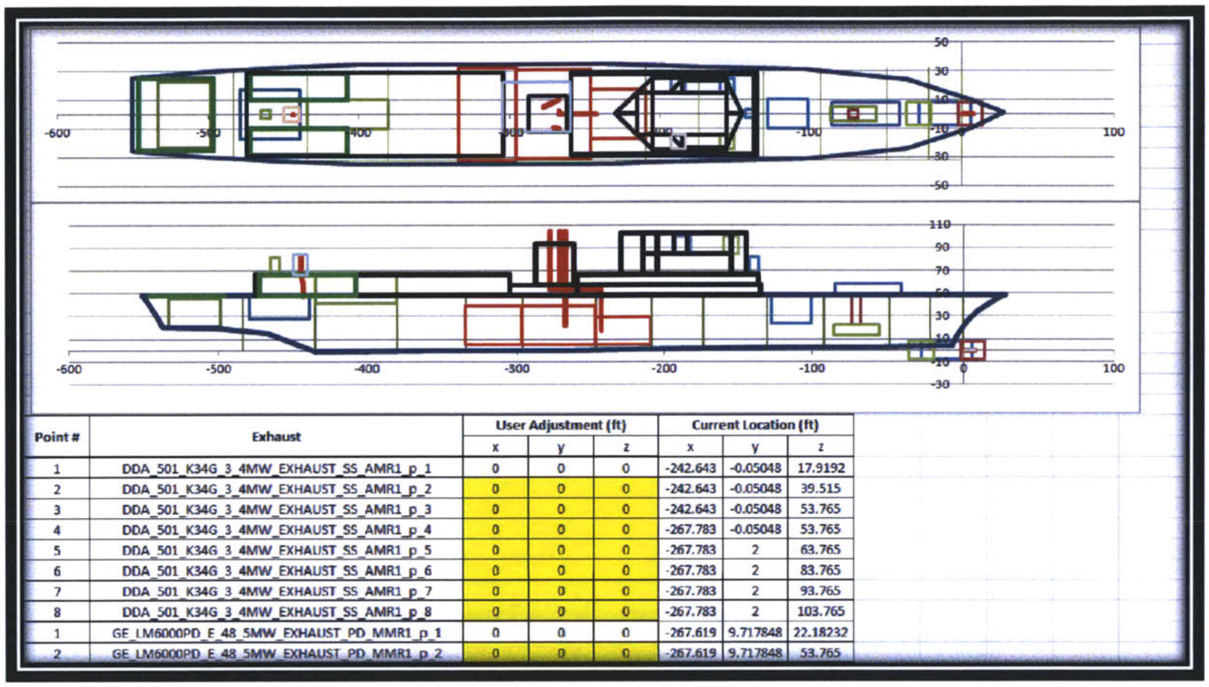
Figure 10: Shape Reference for Deckhouse Construction



block #1		
Shape	rectangle	
	length	angle
side	(ft)	(deg)
1	60	
2	60	
3	60	
4	60	
Bottom Level	Weather_Deck	
Top Level	01	
Long. Location (ft from FP)	150	
Trans Location (ft to PORT)	0	
Name	Officer_Quarters	

Figure 11: Input for Design Block of Deckhouse

The initial layout of exhaust stacks is imported when the user selects an engine plant arrangement from the IPSDM. However, once selected and combined with the afore mentioned deckhouse construction, the ESSDT allows the user the capability to alter exhaust stack paths. The "Adjust_Stacks" tab, illustrated in Figure 12, allows the user to specify an x, y, and z adjustment to each of the guiding points which defines the path of the exhaust stacks.



Point #	Exhaust	User Adjustment (ft)			Current Location (ft)		
		x	y	z	x	y	z
1	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_1	0	0	0	-242.643	-0.05048	17.9192
2	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_2	0	0	0	-242.643	-0.05048	39.515
3	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_3	0	0	0	-242.643	-0.05048	53.765
4	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_4	0	0	0	-267.783	-0.05048	53.765
5	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_5	0	0	0	-267.783	2	63.765
6	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_6	0	0	0	-267.783	2	83.765
7	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_7	0	0	0	-267.783	2	93.765
8	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_8	0	0	0	-267.783	2	103.765
1	GE_LM6000PD_E_48_5MW_EXHAUST_PD_MMR1_p_1	0	0	0	-267.619	9.717848	22.18232
2	GE_LM6000PD_E_48_5MW_EXHAUST_PD_MMR1_p_2	0	0	0	-267.619	9.717848	53.765

Figure 12: Input for adjusting exhaust stack paths

After the major ship's parameters have been selected, major equipment selected, and deckhouse constructed, the program is ready to develop a model in Paramarine. The "Import_to_Paramarine" tab is the final interface. This is shown in Figure 13.

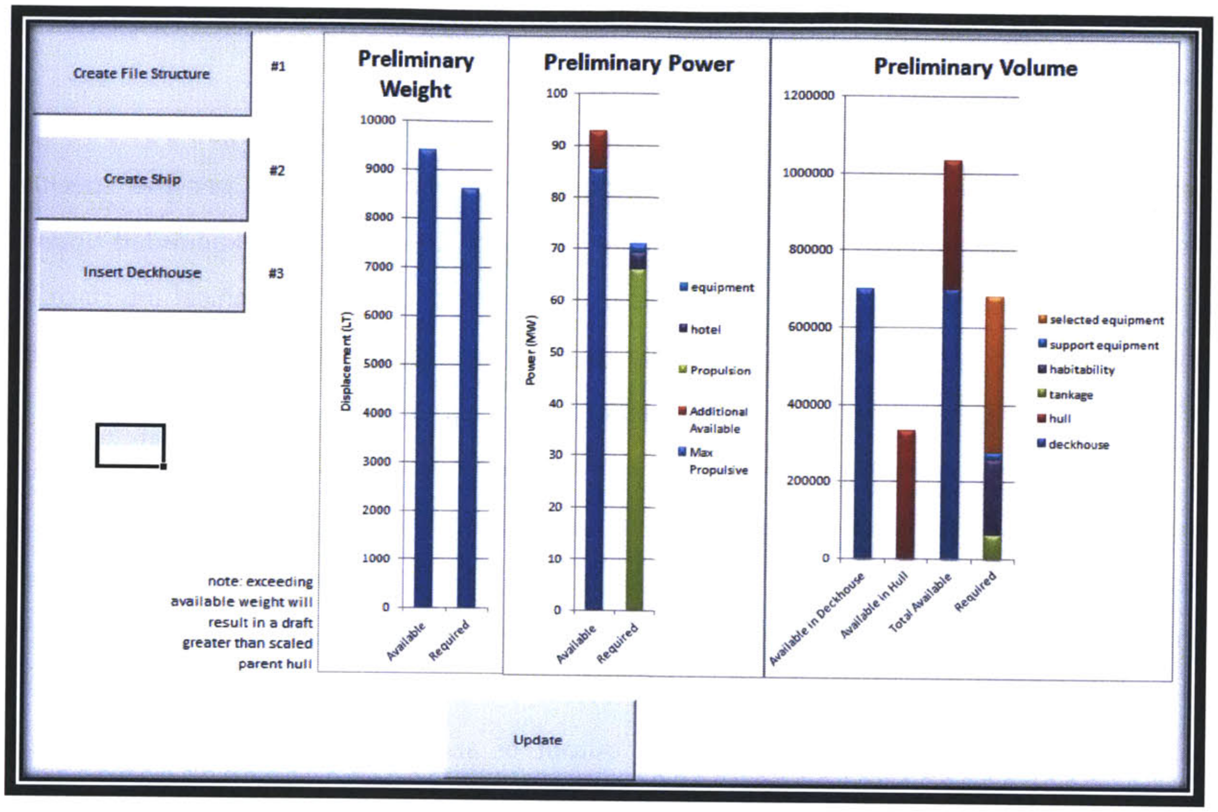


Figure 13: Final User Interface to Export Design to Paramarine

In this tab, the user can view an initial assessment of how the current design's power, volume, and weight available pair up with the selected design. The "Create File Structure" button creates the basic analysis structure required within Paramarine. The "Create Ship" button inserts the selected hull structure along with all major equipment components into Paramarine. After this, the "Insert Deckhouse" option creates a deckhouse within the Paramarine model. Once these three steps have been performed, the user has a preliminary ship design model ready for analysis within the naval architecture program Paramarine.

2.2 User Inputs & Defaults

2.2.1 Initial Input Table

The initial input table, as shown in Figure 6, allows the user to input the ship's parameters of maximum speed, range, installed power, information about the aviation facilities or lack thereof, crew size, store's endurance, and which hull type from those available in the hull library, the user wishes to model.

The maximum speed is, primarily, a component related to the installed power of the selected power plant. In determining the power required to perform per the specified maximum power, the ESSDT makes use of the Admiralty Coefficient, as expounded upon in greater detail in chapter 3. This, in turn, is used as part of the final graphical interface, as shown in Figure 13. It allows the user to visually ascertain if the selected power plant is capable of providing sufficient power to propel the vessel at the desired maximum speed.

The input parameter of "Range" is primarily used in ascertaining the ship's tankage requirements. In conjunction with the endurance speed, this input parameter is used to determine the designed vessel's tankage requirements. This, in turn, goes into the preliminary analysis of volume and weight requirements for tankage allowance for the design.

The selection for "Installed Power" provides the user with a drop down selection of available power plants produced from the output of an Integrated Power System Design Module (IPSDM) produced through research conducted in parallel to this thesis by David Jurkiewicz, a fellow 2012 Naval Engineer's degree candidate, in his thesis *Modular Machinery Arrangement and Its Impact in Early-Stage Naval Electric Ship Design* (Jurkiewicz, 2012).

Beam and depth are driven by equipment and machinery selection. These are used in conjunction with the hull selected to determine initial length and draft of the ship. These values are then compared against default constraints. These constraints may force a design to have a different displacement than initially desired. To alert a user to this potential discrepancy, a "Final Displacement" read-out is provided. This displays the actual resulting displacement as currently determined by the program based on all present user input. The user also has the ability to manually alter the beam or length, but there is a warning provided against this action as it can have detrimental effects on the hull efficiency as compared to the parent hull the resulting ship model is based upon.

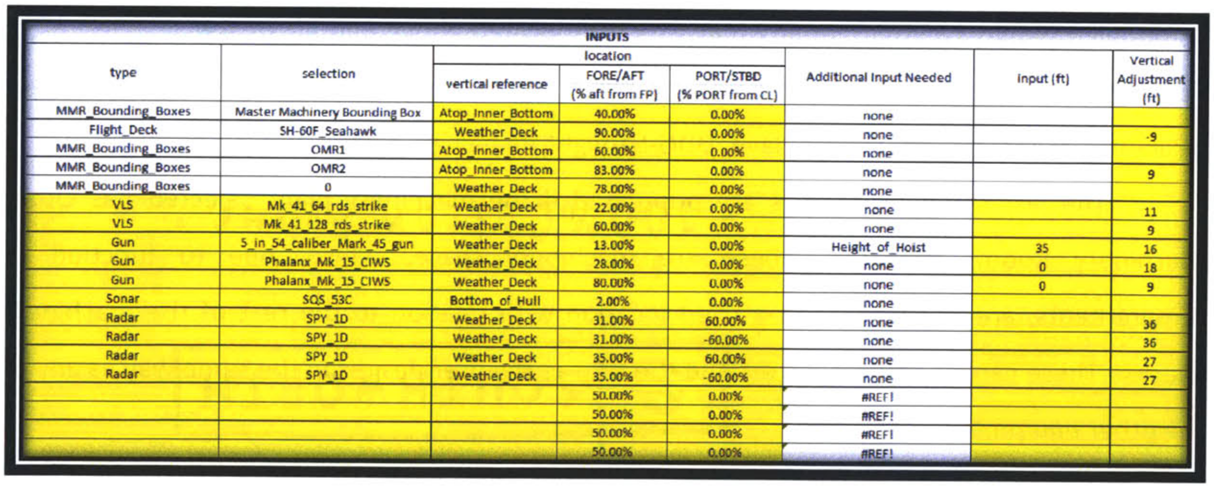
At this point in the process, the user may select whether or not to include aviation facilities in their design. If selected, the user may either have no hangar facilities, one hangar, or two hangars. If two hangars are selected, they may be oriented either fore-and-aft or side-by-side. A helicopter type may also be selected at this time. Based upon this selection, the hangars will be sized accordingly. Placement and sizing of the flight deck is done along with equipment selection. Location and spacing between the hangars can be performed when constructing the deckhouse.

Since the development of IPS has not altered habitability requirements, "Crew Size" and "Endurance" are both used in a fairly traditional manner in conjunction with defaults and historically based parametric equations, to determine the volume and weight requirements which the number of personnel place on the design. Chapter 3 develops this in greater detail.

"Hull Type" allows the user to select a hull from a drop down menu consisting of hulls available in a "hull library". Each of the available hulls has control point information available which allows the program to construct a model of it within Paramarine which has been scaled in accordance with the needs of the designed vessel.

2.2.2 Initial Equipment Selection & Bulkhead Placement

Figure 14 illustrates the input table for equipment selection. The first column allows a drop-down selection of equipment type. Available choices are Gun, VLS, Sonar, Radar, and Misc (for miscellaneous, user defined equipment within the library).



INPUTS							
type	selection	location			Additional Input Needed	Input (ft)	Vertical Adjustment (ft)
		vertical reference	FORE/AFT (% aft from FP)	PORT/STBD (% PORT from CL)			
MMR_Bounding_Boxes	Master Machinery Bounding Box	Atop_Inner_Bottom	40.00%	0.00%	none		
Flight_Deck	SH-60F_Seahawk	Weather_Deck	90.00%	0.00%	none		-9
MMR_Bounding_Boxes	OMR1	Atop_Inner_Bottom	60.00%	0.00%	none		
MMR_Bounding_Boxes	OMR2	Atop_Inner_Bottom	83.00%	0.00%	none		9
MMR_Bounding_Boxes	0	Weather_Deck	78.00%	0.00%	none		
VLS	Mk_41_64_rds_strike	Weather_Deck	22.00%	0.00%	none		11
VLS	Mk_41_128_rds_strike	Weather_Deck	60.00%	0.00%	none		9
Gun	5_in_54_caliber_Mark_45_gun	Weather_Deck	13.00%	0.00%	Height_of_Hoist	35	16
Gun	Phalanx_Mk_15_CIWS	Weather_Deck	28.00%	0.00%	none	0	18
Gun	Phalanx_Mk_15_CIWS	Weather_Deck	80.00%	0.00%	none	0	9
Sonar	SQS_53C	Bottom_of_Hull	2.00%	0.00%	none		
Radar	SPY_1D	Weather_Deck	31.00%	60.00%	none		36
Radar	SPY_1D	Weather_Deck	31.00%	-60.00%	none		36
Radar	SPY_1D	Weather_Deck	35.00%	60.00%	none		27
Radar	SPY_1D	Weather_Deck	35.00%	-60.00%	none		27
			50.00%	0.00%	#REF!		
			50.00%	0.00%	#REF!		
			50.00%	0.00%	#REF!		
			50.00%	0.00%	#REF!		

Figure 14: Table for Initial Equipment Selection

Once an equipment type is selected, a dropdown menu is then available in the next column for the equipment selection. This dropdown menu is populated by all equipment available in the program's library as further discussed in section 2.3 Equipment Library. The user can then place this equipment longitudinally, transversely, and vertically as a proportion of ship's length, a proportion of distance from ship's centerline, and vertical deck reference respectively. In the last column, a further vertical adjustment is allowed to vary location away from the limited number of reference decks available. Reference decks only include "Bottom_of_Hull", Atop_Inner_Bottom", and 'Weather_Deck". This ability to adjust the vertical reference allows all possible placements for any particular equipment item.

The first couple of rows within this input have fixed items. The first row is for placement of the main engine room module as provided by the IPSDM. It allows the user to move the Main Machinery Rooms (MMRs) and Auxiliary Machinery Rooms (AMRs) together within the ship the same way as any other equipment.

The next row allows placement of the flight deck if aviation facilities were selected. The flight deck is sized according to the helicopter selected based upon information stored within the equipment library. It is expected that the standard location for the flight deck will be the weather deck. However, if a vertical displacement of the flight deck is selected, when the ship is modeled within Paramarine, any hull structure located above and aft of the selected flight deck will be removed from the model. This is to allow modeling of stepped flight deck structures as certain ship classes have, DDG-51 flight IIA being a particular example.

The last three fixed rows are populated if the machinery plant selected has Other Machinery Rooms (OMRs). These are machinery spaces which, due to function or requirements, are not located in a fixed location with relation to the rest of the machinery spaces. These can also be moved around the ship as the user desires in the same way as any of the other equipment.

As each of these equipment items is selected and placed, a graphical representation is displayed in the upper right-hand corner of the display as shown in Figure 7.

2.2.3 Hanger & Deckhouse Construction

The tab "Create_Deckhouse" allows for the placement of hangar facilities and the construction of a ship's deckhouse as illustrated in Figure 8. Similar to other equipment components, the hangars are placed as a percentage of ship's length aft from the forward perpendicular. Transverse location is handled in a similar manner. Vertical location is referenced from one of the three aforementioned vertical references, "Weather_Deck", "Atop_Inner_Bottom", and "Bottom_of_Hull". Just as with the other equipment components, a vertical displacement is also allowed for the hangars. This allows variation as can be seen in a DDG-51 Flight IIA where the flight deck and hangar facilities are located one deck below the weather deck with a "cut down" in the hull structure as discussed when dealing with the placement of the flight deck. It should be noted that the flight deck determines what portion of the hull is "cut-out" to allow for aviation operations.

Another input for the hangar facilities is whether or not to count them toward the deckhouse volume. This is an accounting device. If the hangar is considered within the deckhouse, then that volume is removed from the available volume for the deckhouse within the volume bank, as discussed in greater detail in section 3.2.3 Volume Bank.

As previously mentioned, the deckhouse is constructed as a selection of geometric modules consisting of rectangles and triangles. Originally, an interface allowing for control points defining up to eight sides for a module was attempted. This would have been similar to the deckhouse construction utilized in the ASSET version 6 and allowed much greater flexibility in deckhouse construction. However, given the methods Paramarine uses for defining the building blocks which make up the deckhouse, primarily cuboid in nature, this method proved untenably awkward. The simpler selection of only squares and triangles allows for the construction of, effectively, any geometry imaginable. At the same time this constrains the user to input this information in a manner in which Paramarine is able to easily handle it.

As mentioned, Paramarine primarily makes use of basic geometry, most notably cuboid structures, to define its construction geometry for ship models. The triangular structures were able to be modeled by constructing, within Paramarine, a cutting block of greater dimensions then the block being defined. This cutting block was then rotated in such a way that, by subtracting its structure from the module being modeled, the module resulted in the desired triangular shape.

For future iterations of this program, a similar technique could be used to model angled deckhouse structures. A cutting block of sufficient dimensions could be constructed, rotated, and used to subtract from the deckhouse block to create this angled structure. To facilitate this future development, an input for angled sides has been provided within the deckhouse module construction input. At this time, however, this input is not linked to any process; and inputting any information into this file will have no effect on the modeled ship within Paramarine.

As these deckhouse modules are constructed, they are visualized in the image in the upper left of the input page shown in Figure 8. An example can be seen modeled as the black structures within Figure 15. As with all other inputs, though, these modules are not visible within the image depicted in Figure 15 until the Excel file has been updated by one of the previously mentioned methods.

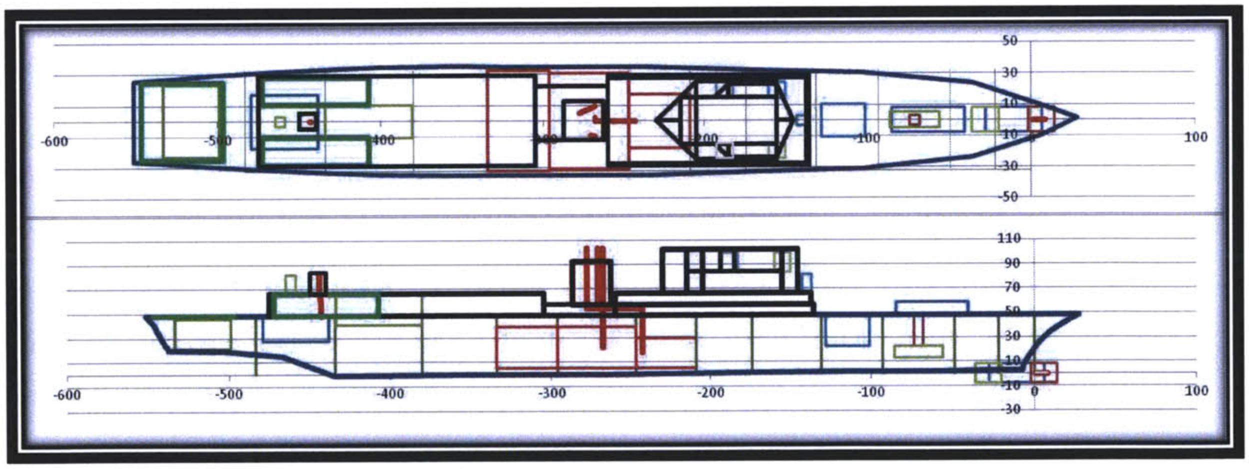


Figure 15: Deckhouse Figure Output

2.3 Equipment Library

The equipment library contains all pre-populated equipment which may be inserted into the design. It is constructed in such a way that more equipment may be included within the library at a future date. This library is referenced by the ESSDT for initial equipment which the user has the option of including with the design, construction of diagrams within Excel, and eventual export into Paramarine. As part of the export process into Paramarine, an equipment library is populated within the Paramarine model with all selected items. This allows a user to further develop the model within Paramarine after initial import. At this time, only the items selected for initial inclusion in the model are imported into Paramarine. However, a future macro within the Excel model could be developed to allow for expanded populating of the equipment library in Paramarine to further facilitate model development once the design process has moved from ESSDT to Paramarine.

There is also a library for hull forms which will be discussed in greater detail in section 3.3 Hull Creation.

2.3.1 Library Components

Equipment stored within the equipment library is categorized as either a gun, Vertical Launch System (VLS), radar, sonar, hangar, flight deck, or miscellaneous. Table 1 and Table 2 contain an abbreviated list of some of the equipment initially included in the ESSDT's library for preliminary testing.

Table 1: Equipment within Library

Gun	VLS	Radar
5_in_54_caliber_Mark_45_gun	Mk_41_8_rds_strike	AN/SPS_49(V)1
Phalanx_Mk_15_CIWS	Mk_41_64_rds_strike	AN/SPS_49(V)2_(V)1
	Mk_41_128_rds_strike	AN/SPS_49(V)3_(V)1
		AN/SPS_49(V)8 ANX_(V)8
		AN/SPS_49A(V)1
		Herakles
		SPY_1D

Table 2: Equipment within Library (cont.)

Sonar	Hanger & Flight Deck	Misc
SQS_53C	SH-60F_Seahawk	none
	Firescouts	
	MH-53E_Sea Dragon	
	MH-60S_Seahawk	
	HH-60H_Seahawk	
	CH-53E_Super_Stallion	

Although there are, currently, no items within the miscellaneous category, it is included for expansion to include future systems which do not easily fall under any of the other categories.

2.3.2 Information Contained Within the Library

Each item contained within the library has entries for all pertinent data for preliminary modeling. For the gun, radar, sonar, and VLS systems, these include the weight with relative centroid, power, clearance box, file locations for graphics, number of people required for operation and maintenance, cooling requirements, electrical requirements, and an option for additional input. The flight deck and hangars include only geometric requirements since the weight of structure would be accounted for in the ship's estimate of Ship's Work Breakdown Structure (SWBS) group 1. However, they do include the ability to enter an additional weight should the requirements for support of a particular type of helicopter require some additional structure such as a reinforced flight deck for a heavy air frame.

For the weight of each item, the library can accommodate up to four different weights with their respective centroids. These can be flexibly used to model the equipment item as deemed best by the user. For example, with a gun, one weight might be used for the gun and mount itself, one to model the hoist system, and one to model the magazine. It is up to the user to decide how to split up the item's weight. The centroids are referenced to a control point which is the point the user places when selecting and placing equipment. For example, a logical control point for a VLS system would be at the center, top of the system where it would be located on the weather deck. If the center of mass for this item is 15 feet below the top of the VLS, the weight's centroid should be entered as -15 ft. Then, when exported to Paramarine, the ESSDT will place the center of mass for the item 15 feet below wherever the user specified the equipment to be placed.

Similar to weight, the library can also store up to four clearance boxes and graphics locations. The clearance boxes are meant to represent the area of and around the equipment item which must be dedicated to that system. Just like the weights, they are spatially referenced to the control point. They may be constructed of cuboids, cylinders, or spheres. The graphics are parasolid object files placed within the graphics folder in the ESSDT program. They must consist of a single solid body. When imported into Paramarine, both the clearance box and graphic are imported into Paramarine's equipment library as shown in Figure 16.

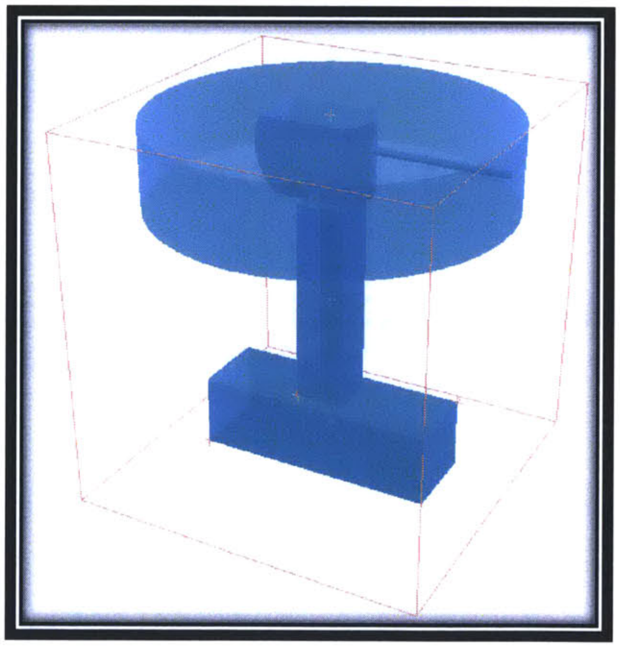


Figure 16: Model of 5" Deck Gun in Paramarine Equipment Library

In Figure 16, the upper cylinder is a clearance box representing the sweep of the gun, the bottom cuboid is a clearance box representing the magazine, and the connecting cuboid is a clearance box representing the hoist system.

However, if there is a graphic for a particular piece of equipment, only the graphic is displayed in the actual ship model. If there is no graphic, the clearance box takes the place of the graphic. This can be seen in Figure 17 where the clearance boxes for the magazine and hoist are displayed since there is no graphic to replace them, but the clearance box representing the sweep of the gun is removed in favor of the graphic designated to replace it.

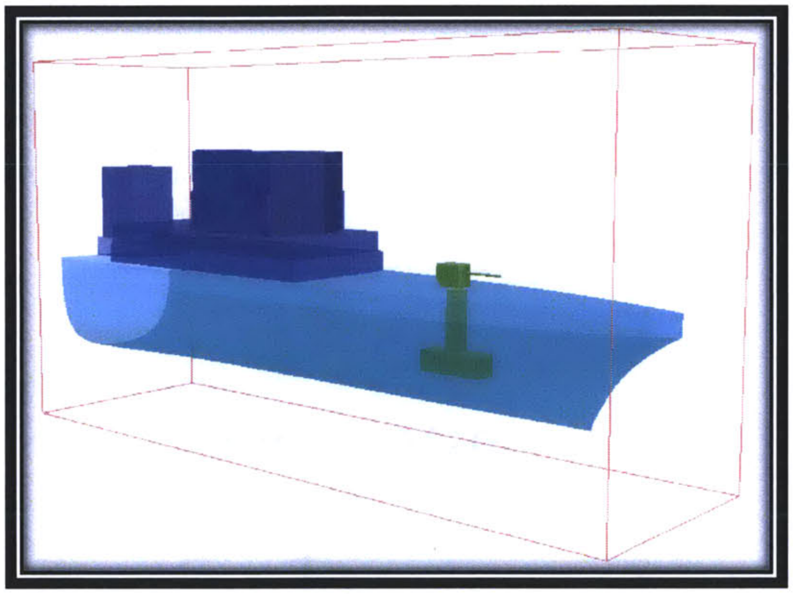


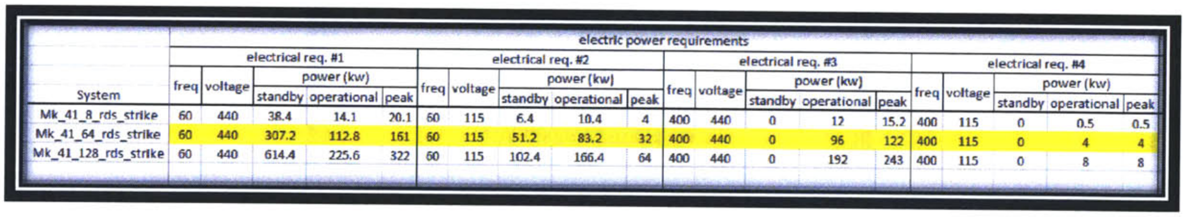
Figure 17: Model of 5" Deck Gun in Paramarine Model

Each of the equipment items also has the ability to enter the number of crew members that an item would require for operation, upkeep, and maintenance. Effectively, it is meant to represent the increase in crew size required by including that particular equipment item. Currently, this entry has no effect on the model. It has been included so that future iterations of the ESSDT might be able to project necessary crew compliments based upon equipment selected rather than just having crew size as a user input.

The cooling requirements have several potential inputs. The equipment can have cooling loads for shore, design, cruise, and battle operational conditions. There is also the ability to specify the relative x, y, and z location of the cooling connection; Header (HDR) connection type (starboard, port, both, or cross connect); and, for the cross-connect situation, how the cross-connect is oriented.

The cooling requirements are imported into Paramarine as service demands under the characteristics of the equipment within the equipment library. As such, they are listed as a demand, what type of cooling they require (currently only chill water), and what maximum energy removal they require. The additional information within the library has been included in a format similar to that required by a similar, previously developed Cooling System Design Tool (CSDT) (Fiedel, 2011). Hopefully, the inclusion of this data for the equipment in this format will allow for eventual incorporation of the CSDT into the ESSDT.

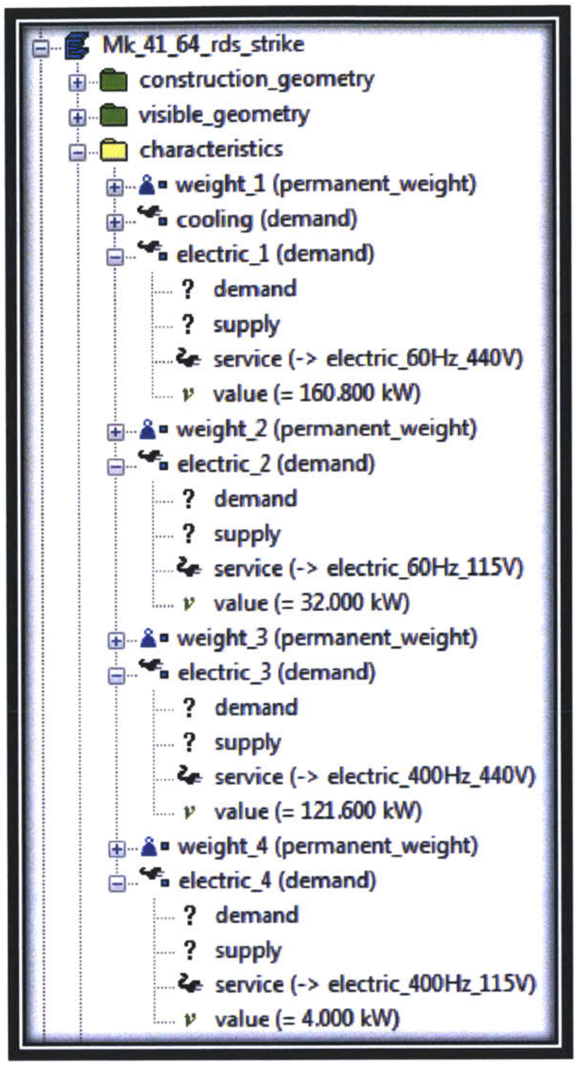
Similar to weight, clearance boxes, and graphics, electrical requirements of the equipment also have four potential loads listed within the database as shown in Figure 18.



	electric power requirements																			
	electrical req. #1					electrical req. #2					electrical req. #3					electrical req. #4				
	freq	voltage	power (kw)			freq	voltage	power (kw)			freq	voltage	power (kw)			freq	voltage	power (kw)		
System			standby	operational	peak			standby	operational	peak			standby	operational	peak			standby	operational	peak
Mk_41_8_rds_strike	60	440	38.4	14.1	20.1	60	115	6.4	10.4	4	400	440	0	12	15.2	400	115	0	0.5	0.5
Mk_41_64_rds_strike	60	440	307.2	112.8	161	60	115	51.2	83.2	32	400	440	0	96	122	400	115	0	4	4
Mk_41_128_rds_strike	60	440	614.4	225.6	322	60	115	102.4	166.4	64	400	440	0	192	243	400	115	0	8	8

Figure 18: Electrical Requirements within Equipment Library

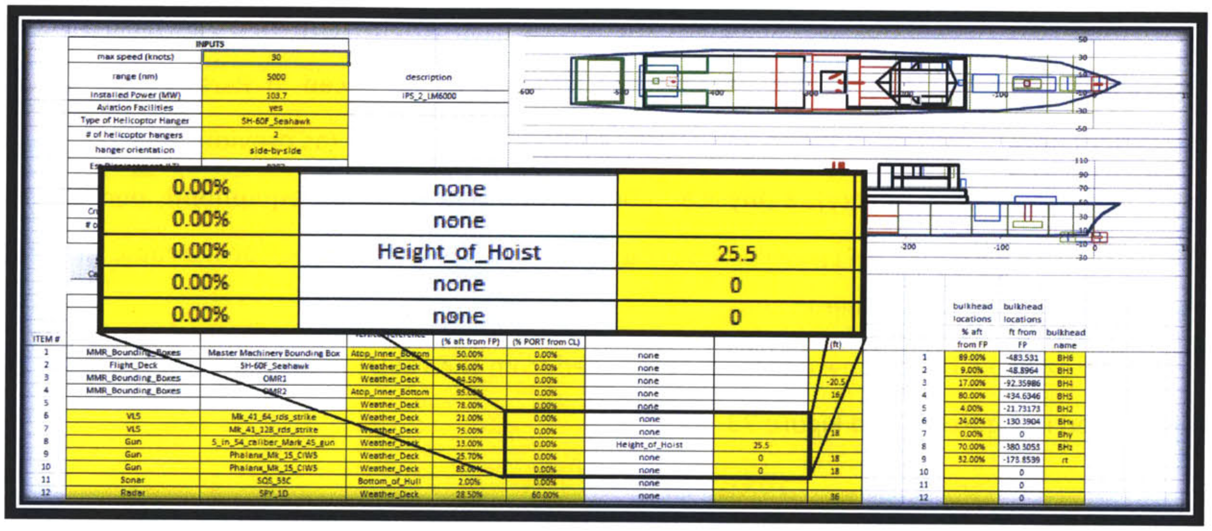
The library provides for a drop down selection of available frequencies (or DC) as well as required voltage and standby, operational, and peak powers. Although not currently utilized, the library also allows the specification of x. y, and z locations for each of these required services. This can potentially allow for the future development of service highways within the Paramarine model. The highlighted information from Figure 18 translates into Paramarine's equipment library as shown in Figure 19.



```
Mk_41_64_rds_strike
    construction_geometry
    visible_geometry
    characteristics
        weight_1 (permanent_weight)
        cooling (demand)
        electric_1 (demand)
            ? demand
            ? supply
            service (-> electric_60Hz_440V)
            value (= 160.800 kW)
        weight_2 (permanent_weight)
        electric_2 (demand)
            ? demand
            ? supply
            service (-> electric_60Hz_115V)
            value (= 32.000 kW)
        weight_3 (permanent_weight)
        electric_3 (demand)
            ? demand
            ? supply
            service (-> electric_400Hz_440V)
            value (= 121.600 kW)
        weight_4 (permanent_weight)
        electric_4 (demand)
            ? demand
            ? supply
            service (-> electric_400Hz_115V)
            value (= 4.000 kW)
```

Figure 19: Equipment's Electric Requirements Inserted in to Paramarine Equipment Library

In addition to the standard equipment data, the library also allows for the addition of one piece of user changeable information. As currently setup, this information can take the form of any one of a clearance box's principle dimensions and any corresponding clearance box centroid that would be altered in one dimension. For example, in the case of a gun with a magazine and hoist, it would be desired to allow a user to change the hoist height for different ships' configurations. To accomplish this, in place of the value for the height of the clearance box, the text "ref" would be entered. It would also be entered into the z centroid for that object and up to one other object which the user wishes to have moved in conjunction with the variable height. The program assumes that the clearance box with both the variable height and z-centroid and the clearance box with only a variable z-centroid are adjacent to one another. The library inputs also allow for a "prompt question" to be entered. This is the question that will appear in the equipment selection table as seen in Figure 20, which illustrates "Height_of_Hoist", and allows the user to input a value.



INPUTS		
max speed (knots)	30	
range (nm)	5000	description
Installed Power (MW)	103.7	IPS_2_LM6000
Aviation Facilities	yes	
Type of Helicoptor Hanger	SH-60F_Seahawk	
# of helicoptor hangers	2	
hanger orientation	side-by-side	

0.00%	none	
0.00%	none	
0.00%	Height_of_Hoist	25.5
0.00%	none	0
0.00%	none	0

ITEM #				(% aft from FP)	(% PORT from CL)			(ft)
1	MMR_Bounding_Boxes	Master Machinery Bounding Box	Atop_Inner_Bottom	50.00%	0.00%	none		
2	Flight_Deck	SH-60F_Seahawk	Weather_Deck	96.00%	0.00%	none		
3	MMR_Bounding_Boxes	OMR1	Weather_Deck	84.50%	0.00%	none		-20.5
4	MMR_Bounding_Boxes	OMR2	Atop_Inner_Bottom	95.00%	0.00%	none		16
5			Weather_Deck	78.00%	0.00%	none		
6	VLS	Mk_41_64_rds_strike	Weather_Deck	21.00%	0.00%	none		
7	VLS	Mk_41_128_rds_strike	Weather_Deck	75.00%	0.00%	none		18
8	Gun	5_in_54_caliber_Mark_45_gun	Weather_Deck	13.00%	0.00%	Height_of_Hoist	25.5	
9	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	25.70%	0.00%	none	0	18
10	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	85.00%	0.00%	none	0	18
11	Sonar	SQS_53C	Bottom_of_Hull	2.00%	0.00%	none		
12	Radar	SPY_1D	Weather_Deck	28.50%	60.00%	none		36

	bulkhead locations % aft from FP	bulkhead locations ft from FP	bulkhead name
1	89.00%	-483.531	BH6
2	9.00%	-48.8964	BH3
3	17.00%	-92.35986	BH4
4	80.00%	-434.6346	BH5
5	4.00%	-21.73173	BH2
6	24.00%	-130.3904	BHx
7	0.00%	0	Bhy
8	70.00%	-380.3053	BHz
9	32.00%	-173.8539	rt
10		0	
11		0	
12		0	

Figure 20: Example of Additional User Selection for Equipment

In addition to the provided library selections of gun, VLS, radar, sonar, hangar, and flight deck, the ESSDT library also has a miscellaneous tab. Although not currently populated, items in this sub-library have all the same standard inputs as the previously mentioned equipment items.

It should also be mentioned that most of the data currently entered in the library is based upon conjecture and estimation for the purpose of testing the program's operation and to avoid inadvertent inclusion of distribution sensitive material. In order to fully meet its potential, the ESSDT should be populated with actual equipment data and expanded with a larger variety of equipment to truly model available configurations allowed with today's level of development.

2.4 Product Produced in Paramarine

The real strength of the ESSDT as a tool comes from its ability to leverage the analysis capabilities of Paramarine. Paramarine is a highly versatile ship design analysis tool. A large part of that capability comes from its flexibility in creating the methods behind developing and linking the different aspects of a model. Unfortunately, it is this very flexibility which can make Paramarine somewhat unwieldy.

The ESSDT simplifies user selection and leverages an equipment library to facilitate the user construct of a model within Paramarine. The model produced in Paramarine has an equipment library with all included equipment and their parameters (as previously discussed in section 2.3 Equipment Library); a physical model of the ship with hull, equipment, deckhouse, stacks, and helicopters; initial power vs. speed curve; and the ability to perform a rapid first-cut assessment of trim and stability.

An example of the physical model produced from the ESSDT input modeled in Figure 21 can be seen in Figure 22 and Figure 23.

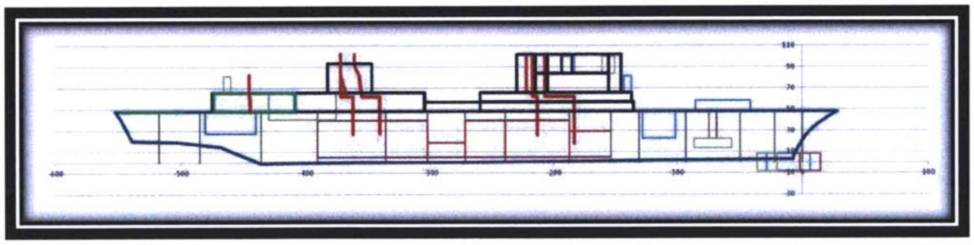


Figure 21: Example Output Model from the ESSDT

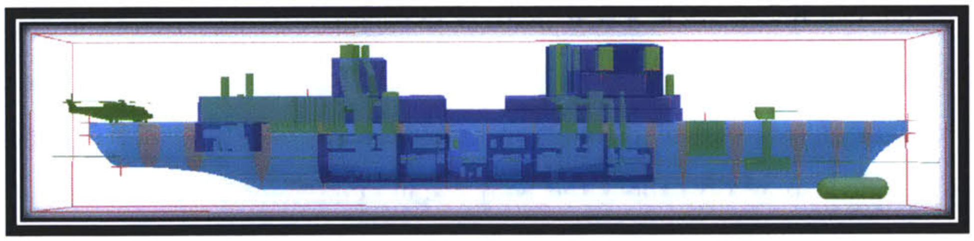


Figure 22: Example of Physical Model Produced in Paramarine, Profile View

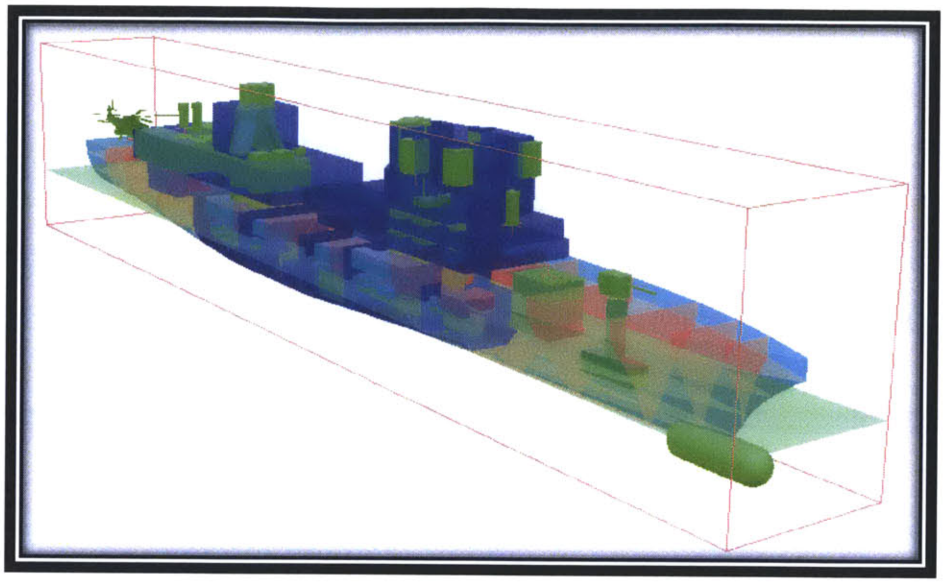
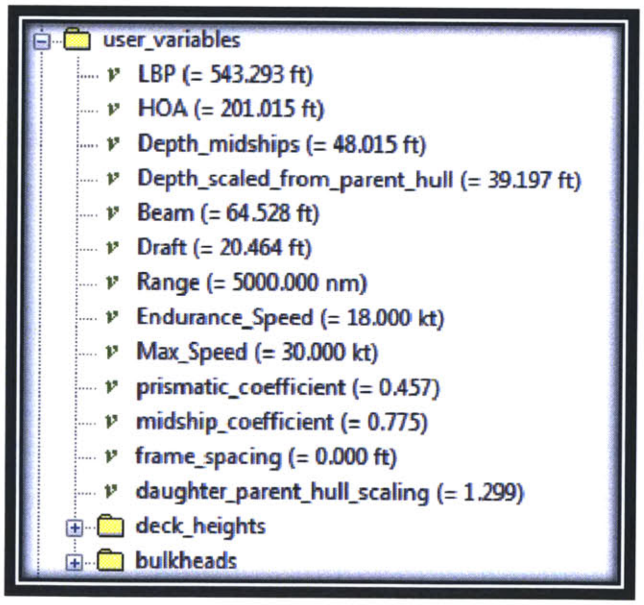


Figure 23: Example of Physical Model Produced in Paramarine, Perspective View

This model includes all equipment, including all of the equipment from the selected engine plant which would have been previously modeled in the IPSDM, hull, bulkheads, decks, and stacks. Additionally, all of these physically modeled items are linked to a "user variable" folder which allows rapid alteration of principle ship's dimensions and factors as can be seen in Figure 24.



```
user_variables
    LBP (= 543.293 ft)
    HOA (= 201.015 ft)
    Depth_midships (= 48.015 ft)
    Depth_scaled_from_parent_hull (= 39.197 ft)
    Beam (= 64.528 ft)
    Draft (= 20.464 ft)
    Range (= 5000.000 nm)
    Endurance_Speed (= 18.000 kt)
    Max_Speed (= 30.000 kt)
    prismatic_coefficient (= 0.457)
    midship_coefficient (= 0.775)
    frame_spacing (= 0.000 ft)
    daughter_parent_hull_scaling (= 1.299)
  deck_heights
  bulkheads
```

Figure 24: User Variable Folder

The equipment, itself, is located under the principle design building block folder within the design folder and is linked in the typical fashion for Paramarine modeling. This allows a user familiar with Paramarine to make changes to the set-up at will.

The model produced in Paramarine is also linked in such a way as to allow the estimate of a powering curve. An example of this type of analysis can be seen in Figure 25.

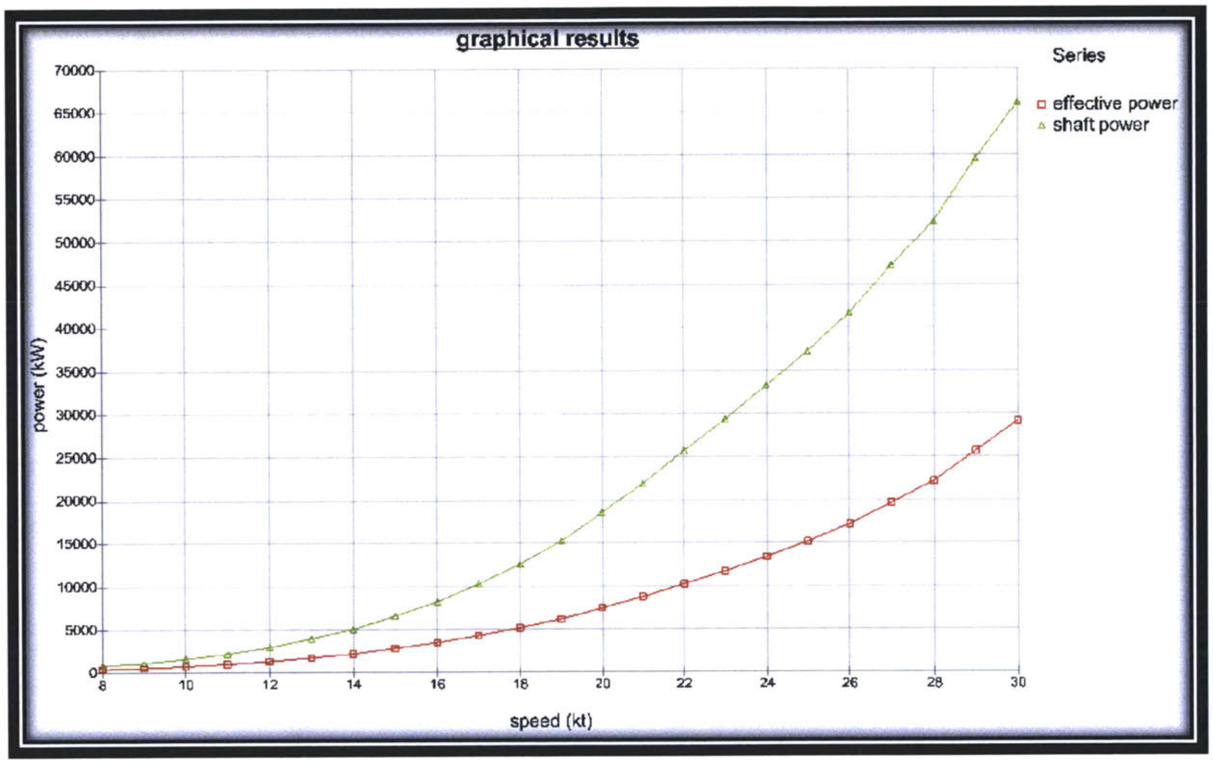


Figure 25: Example of Powering Curve Output from Paramarine

The produced model also allows a preliminary analysis of the model's trim, stability, and GZ curve. Unfortunately, the linkages required within Paramarine to perform this analysis have not been completely automated, and there are some linkages which have to be made manually to allow Paramarine to derive this stability analysis. The exact linkages needed and how to perform them are covered in

Appendix A: Program Instruction Manual.

In addition to the previously mentioned analyses, the model is inserted into the Paramarine file structure with the requisite cross-links to further facilitate expedited linkage and completion of the more in depth analysis afforded by Paramarine. In short, the ESSDT leaves the user with a rapidly constructed Paramarine model with links which might take several days to construct and interlink by manual means. Further, potential analysis is discussed in 5.2 Areas of Future Study & Development.

2.5 Potential Applications

. Given the nature of the ESSDT, there are two primary potential applications. The first would be as a tool for rapid comparison between ships, potentially as part of a process to populate a trade-space with a myriad of designs. The second application would be to facilitate a naval architect or engineer in rapidly synthesizing a base-line model to begin further development within Paramarine.

The ESSDT is set up in such a way as to allow the construction of a base-line ship. Once a base-line is established, one or two parameters, such as selected engine plant or different radar systems, can be quickly switched out. Then a model for in-depth analysis can be populated within Paramarine. This means that a designer can quickly and simplistically arrive at multiple models in a program capable of fairly flexible, rigorous analysis. An example of this approach can be found in the second case study in section 4.2 Trade-off Comparison of Different Sized Power Plants. If enough of these designs, with variations around a central design, are developed, a designer can garner an appreciation for the way a design space may be populated. This methodology could also be used in conjunction with a Design of Experiments (DOE) or any other early stage design-narrowing methodology.

These early stage designs also provide a good "jumping-off point" for more in-depth design. Because of Paramarine's flexibility and power, it requires a significant amount of time to structure and populate the design tree within Paramarine to enable any kind of significant analysis. The ESSDT automates a good portion of this preliminary set-up. This enables a designer to rapidly build the foundation within Paramarine upon which a more fleshed-out design can be developed. In effect, it gives the designer the ability to get a running start on a design. This ability to rapidly model early stage design empowers the ESSDT to be used for design space population or initial modeling in a fairly flexible manner.

3.0 Design Tool Concepts

While Section 2.0 Use of the Design Tool highlights the general process and attributes of the ESSDT, this section discusses the underlying calculations, parametrics, and assumptions the program uses behind the scenes.

3.1 Excel Interface

In addition to the input tabs provided by the Excel interface as previously discussed, the Excel interface also provides the platform for the preliminary calculations in the ship design process. It accomplishes this by combining the user-entered ship's specifics, provided default values, and some standard naval vessel parametrics. It then uses these calculations to provide both the requisite outputs to the Paramarine model as well as visual guidance within the Excel interface in the form of a ship's profile and layout sketches and graphical representation of required vs. available power, volume, and weight.

3.1.1 Defaults

A complete list of all employed defaults can be seen in Appendix B: ESSDT Entry, Analysis, and Output Worksheets. In general, the default values fall into three broad categories: margins, allowances, and efficiencies; parametric constants; and ship geometry constraints. The goal of the defaults was to limit the number of inputs a user would have to manipulate to assist in rapid modeling whilst allowing the user developing more advanced or non-standard designs to have the flexibility to make more in-depth use of the ESSDT.

The ESSDT makes use of design margins for power, volume, and weight. These are to account for the uncertainty in the design process this early on. The initial default value for each of these is 10%. This factor is fed into estimates for these values in the bank calculations discussed in greater detail in section 3.2 Bank System. The calculated required values for power, weight, and volume are each increased by this margin to help assure sufficient availability of these items in the design.

Allowances are incorporated to plan for future ship expansion through service life. There are allowances for weight, volume, and power. Similar to the margins, these allowances are added to the required values for weight, volume, and power.

A default transmission efficiency and propulsive efficiencies at both max speed and cruising speed are also included amongst the defaults. These are used to determine approximate tankage volume and maximum propulsive power as discussed later in section 3.2 Bank System.

Two items which are effectively both margins and parametrics are the frame factor and passageway factor. These are subtracted in the determination of available volume. They are used to account for the proportion of volume within the ship taken up by support structure and passageways respectively. In the initial set-up of the ESSDT, they are sent to 5% for frame factor and 10% for passageway factor. These are, currently, placeholder values; and more research may help to refine more specific values based upon historical ship design.

Table 3 shows the parametric defaults, their initial values, and their source. The habitability requirements were increased to account for racks, desk, personal stowage, lavatories, etc. since the habitability manual only specifically accounted for "unobstructed area" (Naval Sea Systems Command, 1995). These defaults were incorporated into the design banks as discussed in greater detail in section 3.2 Bank System.

Table 3: Parametric Defaults

Default	Initial Value	Source
Hotel Load per Person	10 kW	(MIT 2N Program, 2011)
Potable Water per Person	0.149 LTs	(Naval Sea Systems Command, 1995)
Sewage per Person	2 ft^3	(MIT 2N Program, 2011)
KG/Depth for Weight Group 1	61.71%	(Society of Allied Weight Engineers, 2006)
CO/XO Area Required per Person	110 ft^2	(Naval Sea Systems Command, 1995) + 30 ft^2
Officer Area Required per Person	75 ft^2	(Naval Sea Systems Command, 1995) + 30 ft^2
Enlisted Area Required per Person	50 ft^2	(MIT 2N Program, 2011)
Minimum Stores	300 ft^2	(MIT 2N Program, 2011)
Stores Area Density	0.0158 ft^2/lb	(MIT 2N Program, 2011)
Stores per Person	9 lb/day	(MIT 2N Program, 2011)
% of Payload Required for Support Space	5%	(MIT 2N Program, 2011)

In addition to margins and parametrics, a number of defaults are included to assist in the determination of ship's geometry. A complete list is provided in Appendix B: ESSDT Entry, Analysis, and Output Worksheets. They include items such as the heights of decks, structural size approximations, and allowable range for the ship's length to beam ratio. A split point is provided to indicate at what percentage of ship's length placed equipment is turned around to face aft, but this is not currently incorporated into the program. It was left in the defaults in the hope that future iterations would be able to incorporate it. A rough placement of the fuel tankage centroid is also provided. This is used in preliminary stability calculations until the user further defines the design within Paramarine and specifies specific compartments to be tanks.

The use of these defaults are utilized within the calculations of Excel and feed into the bank system and, eventually, the Paramarine model.

3.1.2 Calculations

The Excel program uses the provided defaults in conjunction with the user inputs and library to perform many of the preliminary calculations for the ship. These include ship's dimensions as well as power, weight, and volume required and available. The required and available values will be discussed further in section 3.2 Bank System.

For the basic ship's dimensions, several techniques were attempted. In the final iteration of the program, the beam is determined by the selected equipment and machinery space. Taking the width of all selected components and the percentage of beam that they are offset from the ship's centerline, the program determines the minimum beam to accommodate selected items. With few exceptions, the width of the selected machinery spaces is almost always the driving factor in the calculation of this minimum beam.

Once the beam is determined, it is used in conjunction with the selected hull to determine the rest of the ship's dimensions. The selected hull has information of the parent hull stored in the hull library. This database supplies V_1, the parent hull volumetric displacement, as well as B_1, L_1, D_1, and T_1 the parent hull's beam, length, depth at midship, and draft respectively. Once B_2, the model's beam, is determined as previously discussed, its volumetric displacement can be determined as follows:

$$V_2 = V_1 * \left(\frac{B_2}{B_1}\right)^3$$

Then its length and draft can be determined as follows:

$$L_2 = L_1 * \left(\frac{V_2}{V_1}\right)^{\frac{1}{3}}$$

$$T_2 = T_1 * \left(\frac{V_2}{V_1}\right)^{\frac{1}{3}}$$

This approach allows for uniform scaling of the parent hull to help retain some of the parent hull's hydrodynamic efficiency.

The required midship's depth is defined by D_m, the depth of the machinery space; D_{ib}, the height of the inner bottom; and the number, n, and height of decks, h, within the hull above the machinery space. The equation below determines the minimum required depth.

$$D_2 = D_{ib} + D_m + (n * h)$$

To avoid unequal scaling of the hull in the vertical dimension which could negatively impact hydrodynamic efficiency compared to the parent, the parent hull's depth is scaled in the same manner as the length and draft. If additional depth is needed to accommodate the selected machinery spaces and hull decks, it is added within Paramarine as a vertical walled section as shown in Figure 26.

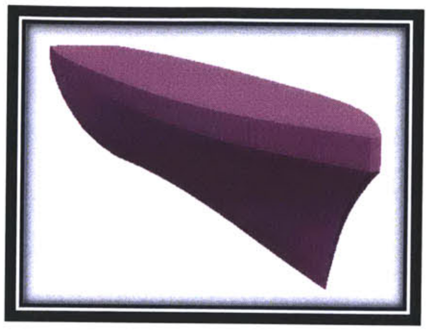


Figure 26: Scaled Hull with Added Hull Addendum

If the scaled hull has a depth greater than that required, the program defaults to the scaled hull.

Although this approach isn't aesthetically ideal, it gives the user a template within Paramarine which can be used to further develop the hull geometry while preserving the hydrodynamic efficiencies of the parent hull. For example, with a minimal amount of manipulation within Paramarine, this structure can be used to produce a tumble home design to allow a reduced RCS, as shown in Figure 27.

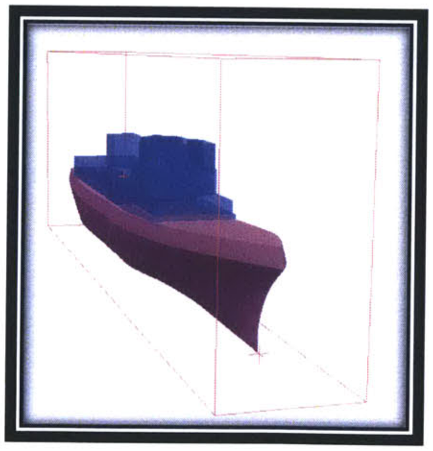


Figure 27: Hull Addendum Transformed into Tumblehome

These calculations are then paired with the location and data for the previously selected equipment, the calculations for the bank system, and the information for the hull to create the model within Paramarine.

3.2 Bank System

The underlying premise behind the bank system in the ESSDT is to provide the user a quick, visual way to track the available and required power, volume, and weight. The visual representation is on the same tab as the buttons allowing export to Paramarine, as can be seen in Figure 13: Final User Interface to Export Design to Paramarine. As an off-shoot of this bank determination, some of the calculations performed for the bank system are also exported into the Paramarine model.

3.2.1 Power Bank

The available power is provided from the output of the selected machinery configuration from the IPSDM. This output is comprised of two principle values, the power supplied and the maximum propulsive power available. Required power is comprised of three parts, propulsive, hotel, and equipment requirements.

The maximum propulsive power available is graphically displayed as the blue portion of left-hand bar as shown in Figure 28. This represents the maximum power which the motors of the selected machinery arrangement can dedicate to propulsive power. The power supplied is represented by the entirety of the bar on the left, both red and blue portions. This represents the entirety of power the selected machinery arrangement can supply to the ship less the plant's own internal needs.

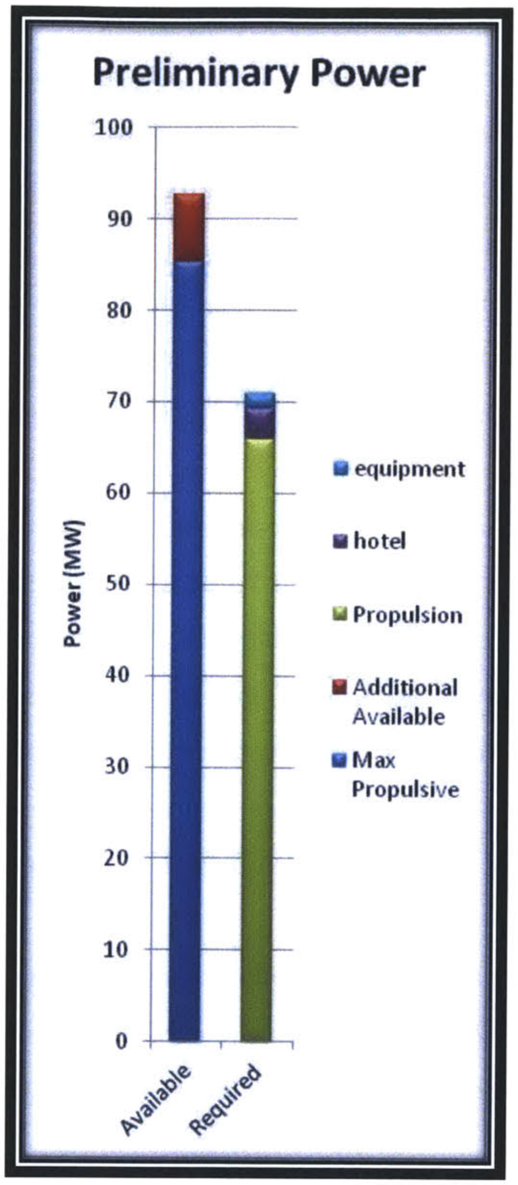


Figure 28: Power Bank Display

Propulsive requirements are determined using the Admiralty Coefficient Method (Stokoe, 2003), the equation for which is:

$$P_{eff} = \frac{\Delta^{\frac{2}{3}} * V^3}{1000 * C}$$

where P_{eff} is the effective propulsive power in MW, Δ is the ship's displacement in long tons, V is the ship's maximum speed in knots and C is the Admiralty Coefficient of the parent hull. This value is then divided by the default values for transmission efficiency and propulsive coefficient at maximum speed to determine the maximum propulsive power needed. This is a first-cut preliminary value for comparison prior to model creation within Paramarine. Once the model is created, the Paramarine file structure is set-up to automatically produce a more detailed power curve for the particular hull modeled.

Hotel load is determined by combining the default power per person with the number of personnel selected by the user. A more detailed analysis in accordance with the guidelines set out in the Navy's Design Data Sheet (DDS) 310-1 was run for comparison (Naval Sea Systems Command, 1988). The more complexly determined value did not differ by more than 10% from the simpler calculation used, and the simpler calculation was typically more conservative. So it was determined that, at this phase of the design process, the simpler method was adequate.

Equipment load is determined by summing the max load for all selected items. Then the propulsive, hotel, and equipment loads are all increased by the default design margin and the hotel and equipment loads are increased by the service life allowance. The propulsive power is not increased by the service life allowance since, in order to have the same maximum speed, it is assumed that any future alteration for propulsion would decrease required power, not increase.

Once these available and required power loads are determined, they are graphically displayed as seen in Figure 28. Of the calculated values, the only one carried over to the Paramarine model is the maximum load of the selected equipment.

3.2.2 Weight Bank

Similar to the power bank, the weight bank displays available and required weight as seen in Figure 29.

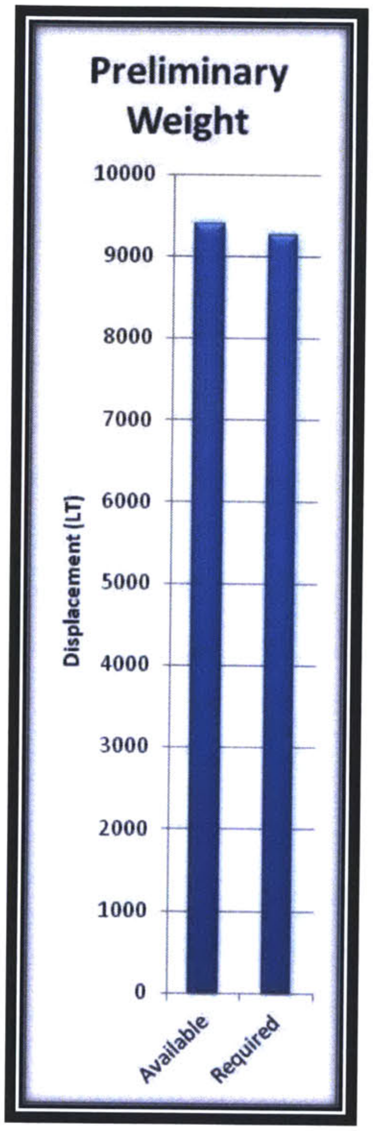


Figure 29: Weight Bank Display

The available weight is purely a matter of hull geometry since the ship's Length Between the Perpendiculars (LBP), beam at the waterline, and draft have all been determined. Therefore, available weight would be:

$$\Delta_2 = L_2 * B_2 * T_2 * C_p * C_m$$

where C_p is the prismatic coefficient and C_m is the midship coefficient.

Determining required weight necessitates the determination of each of the component SWBS group weights. Each is handled individually depending on the individual peculiarities of each group.

Weight group 1, ship's structure, is determined by "smearing" the approximate size of ships support structure into the default hull and deckhouse thicknesses. This is used to determine an approximate volume of steel within the ship's structure which is then multiplied by a default value for the density of steel. To this value is added a parametric equation for the weight of support structures (MIT 2N Program, 2011) as follows:

$$W_{support} = 0.0675W_2 + 0.072(W_3 + W_4 + W_5 + W_7)$$

where W_2, W_3, W_4, W_5, and W_7 are the weights of their respective groups.

Due to the potentially imprecise nature of this method, a different method was used for determination of Weight Group 1 within Paramarine to enable comparison between the two methods if desired by the user. Once in Paramarine, the estimate for the model's weight group 1 is determined using the method put forth by Usher and Dorey in 1982 (Usher & Dorey, 1982) which is a geometric method which scales from envelope dimensions and uses the following equation:

$$W_1 = L_2{}^{2.154} * \left(B_2{}^2 * D_2\right)^{\frac{1}{3}}$$

where the length, beam, and depth are all in meters and W_1 is given in metric tons.

Weight group 2, propulsion systems; weight group 3, electrical systems; and weight group 7, armament, are determined simply by summing the weights of the individual equipment items imported from the IPSDM when the machinery set-up was selected by the user for inclusion within the model. There is no need to export the weight group 2, 3, and 7 estimates to Paramarine since all of the equipment with its respective weights are imported into Paramarine which sums them in its analysis.

Weight group 4, control and communication, consists of four items. The first three are determined by standard parametric equations (MIT 2N Program, 2011). These are the weight for the ship's gyro and navigation equipment, miscellaneous group 4 items, and data cabling. The equations generating these weights in long tons are as follows:

$$W_{gyro\,\&\,nav} = 4.65 * C_{\#}$$

$$W_{misc\ GP\ 4} = 2.24 * C_{\#}$$

$$W_{cabling} = 0.04 * \left(W_{gyro\ \&\ nav} + W_{misc\ GP\ 4} + W_{GP\ 4\ equip}\right)$$

where $C_{\#}$ is the ship's cubic number given by:

$$C_{\#} = \frac{LBP * B * (T + F_{AV})}{10^5}$$

and F_{AV} is an effective average freeboard given by:

$$F_{AV} = \frac{(D_0 - T) + 4(D_{10} - T) + (D_{20} - T)}{6}$$

with:

$W_{GP4equip}$ = Weight of Additional Selected Equipment [LT] (see below)

LBP = Length Between the Perpendiculars [ft]

B = Beam [ft]

T = Draft [ft]

D_0 = Hull Depth at the Forward Perpendicular [ft]

D_{10} = Hull Depth at Midship [ft]

D_{20} = Hull Depth at the Aft Perpendicular [ft]

These three weights are imported into the Paramarine model as individual weights within the design. The fourth weight in group 4, $W_{GP4equip}$, is a summation of additional, selected equipment. This is determined as groups 2, 3, and 7 were, and the result is not exported into Paramarine for the same reason.

For weight groups 5, auxiliary systems, and 6, outfit and furnishings, parametric equations are used to determine the weights of general auxiliary systems, the distiller, and auxiliary system fluids for group 5 and hull fittings and personnel related weights for group 6 using the following equations with results in long tons (MIT 2N Program, 2011):

<u>Weight Group 5</u>

$$W_{gen\ aux\ sys} = \{[0.000772(V_T)^{1.443} + 5.14V_T + 6.19(V_T)^{0.7224} + 377N_T + 2.74P_I] * 10^{-4}\} + 113.8$$

$$W_{distiller} = 0.0013(6.5N_T + 250)$$

$$W_{fluids} = 0.000075V_T$$

<u>Weight Group 6</u>

$$W_{hull\ fittings} = 31.4 + 0.0003187V_T$$

$$W_{personel\ related} = 0.8(N_T - 9.5)$$

where V_T is the total volume of the hull and deckhouse in cubic feet and N_T is the total number of personnel. These weights are then exported to the Paramarine model in the same way the three items of group 4 were.

To account for variable loads, weight for fuel and potable water is also determined. Weight for potable water is based upon the number of crew members and the default requirement (Naval Sea Systems Command, 1995). This value is exported to Paramarine. The weight of fuel is determined by combining the volume of fuel required, which is calculated as part of the volume bank discussed in section 3.2.3 Volume Bank, with a default density of fuel. This value is not exported into Paramarine since the Paramarine model is set-up to independently calculate this weight.

Once all of these requirements have been determined, they are increased by the default values for the weight design margin and the weight service life allowance. This produces a final required weight which is displayed adjacent to the required weight as shown in Figure 29: Weight Bank Display.

3.2.3 Volume Bank

The display for the volume bank, as shown in Figure 30, has two available values, the deckhouse volume and hull volume. The hull volume is determined by taking the parent hull volume and multiplying it by the cube of the ratio of daughter to parent hull beam, similar to how the daughter's volumetric displacement was determined. Since the required hull depth often exceeds the scaled depth, the deck area is multiplied by the difference to account for the additional volume of the hull addendum as shown in Figure 26: Scaled Hull with Added Hull Addendum. When compared to the actual final model's hull volume produced by Paramarine, this method varies by less than 1% from the true hull volume. The deckhouse volume is calculated from geometric calculation of the chosen deckhouse modules. It should be noted that, if the deckhouse modules overlap, the volume of the overlap will be double-counted.

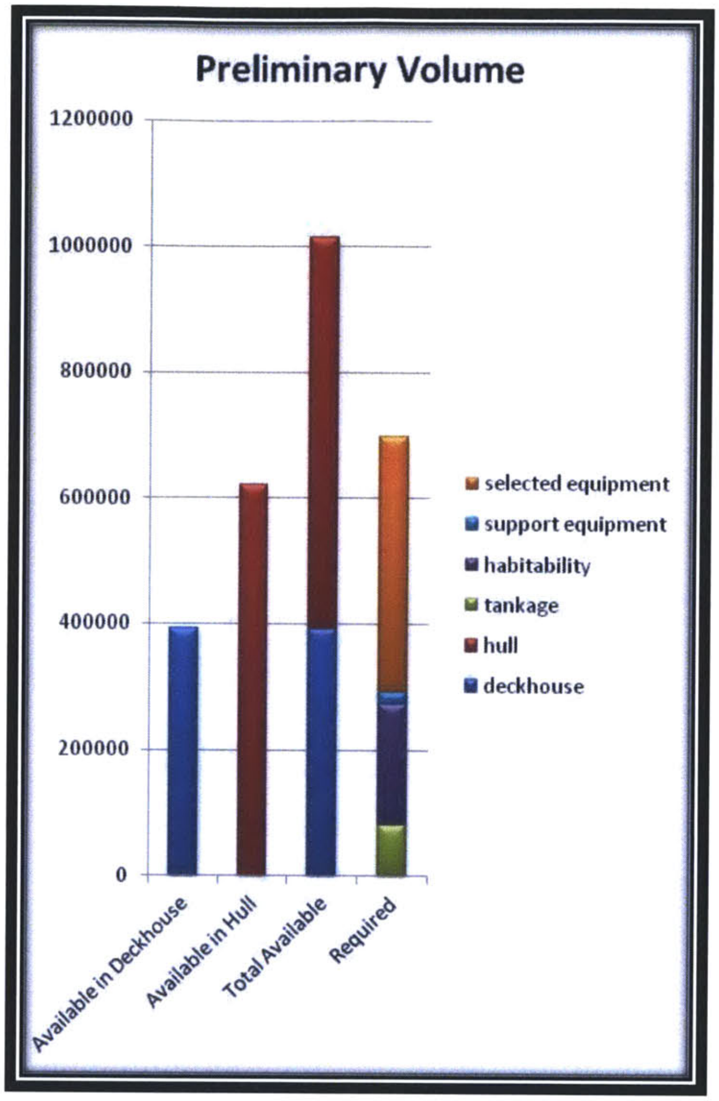


Figure 30: Volume Bank Display

Once the available volume is determined, it is reduced in accordance with the passageway factor and frame factor defaults.

The determination of required hull volume consists of the selected equipment, support equipment, habitability, and tankage. The selected equipment volume is determined by the sum of the volumes of the individual equipment selected and that imported from the IPSDM. Support equipment and habitability are determined by the following parametric equations:

Habitability (Naval Sea Systems Command, 1995)

$$V_{habitability} = (2 * H_{DH} * A_{CO\&XO}) + (N * P_{off} * H_{DH} * A_{off}) + [N * (1 - P_{off}) * H_H * A_{enl}] + V_{stores}$$

with

$$V_{stores} = [A_{\min stores} + (N * \rho_{stores} * W_{stores} * T_{stores*})] * H_H$$

Support Equipment (MIT 2N Program, 2011)

$$V_{maintenance} = P_{support} * \left[\left(2 * H_{DH} * A_{\frac{CO}{XO}}\right) + \left(N * P_{off} * H_{DH} * A_{off}\right) + V_{equipment}\right]$$

&

$$V_{bridge\,\&\,chartroom} = 16ft * (B_2 - 18ft) * H_{DH}$$

where

$V_{habitability}$ is the volume of habitability spaces,

V_{stores} is the volume of ship's stores,

$V_{maintenance}$ is the volume of maintenance equipment,

$V_{equipment}$ is the volume of selected, non-machinery room equipment,

$V_{bridge\,\&\,chartroom}$ is the volume of the bridge and chartroom,

H_{DH} is the average height of a level within the deckhouse,

H_H is the average height of a deck within the hull,

$A_{CO\&XO}$ is the area allocated to the ship's Commanding Officer (CO) and Executive Officer (XO),

A_{off} is the area allocated to the ship's officers,

A_{enl} is the area allocated to enlisted crew,

N is the number of ship's crew,

P_{off} is the percentage of ship's crew consisting of officers,

ρ_{stores} is the area density of ship's stores,

W_{stores} is the weight of stores provided per person per day,

and T_{stores} is the period of time ship's stores is to accommodate.

In addition, the volume required for exhaust and inlet stacks is approximated by multiplying the length of the stacks by their cross-sectional area.

For tankage requirements, the ESSDT estimates fuel tank volume, potable water volume, sewage tankage volume, waste oil volume, and clean ballast volume. For fuel tankage requirements, the ESSDT uses the inputs of range and cruising speed combined with the Specific Fuel Consumption (SFC) for optimal speed provided by the IPSDM to determine the tankage requirements from the power loads of propulsion at cruising speed (from the Admiralty Coefficient as previously discussed), hotel loads, and operational equipment power loads. Potable water requirements are determined from the per person default requirements initially populated from the Navy's Habitability Manual (Naval Sea Systems Command, 1995). Sewage tank volume, waste oil volume, and clean ballast volume are all supplied via the parametric equations below (MIT 2N Program, 2011):

$$V_{sewage} = N * (default\ volume\ of\ sewage\ per\ person)$$

$$V_{waste\ oil} = 0.0005 * (underwater\ hull\ volume)$$

$$V_{Clean\ Ballast} = 0.032 * (underwater\ hull\ volume)$$

None of the volumes calculated within the volume bank are exported to Paramarine. They are only intended to be a general, preliminary guide to the user prior to export to the Paramarine model.

3.3 Hull Creation

Hull creation by the ESSDT in Paramarine is based upon use of a parent hull from a previous design. These hulls are stored within a hull library in the ESSDT along with all pertinent information for hull synthesis within Paramarine as well as requisite guidance for the previously discussed bank assessment.

3.3.1 Paramarine Mechanics of Hull Insertion

Several methods for hull insertion within Paramarine were examined as potential options for the ESSDT to synthesize a hull. The Intelli-Hull Method, which forms a basic hull of a selected type based on general inputs, was deemed to not have the flexibility desired for the ESSDT. At the other end of the spectrum, creating the hull from offsets, as Paramarine allows, seemed too complex a method to easily allow the automation for hull generation desired. Between these two extremes, it was determined that the Quick Hull Method (QHM) afforded greater flexibility than the Intelli-Hull Method while still being simple enough to allow automation of the process.

The QHM process is illustrated in Figure 31. The first item the QHM needs are nine control points illustrated in step one. These points are located at the top and bottom of the bow, transom, forward portion of Parallel Mid Body (PMB) and aft portion of the PMB. There is an additional point located at the beginning of the aft cut-up. The reason the example in Figure 31 appears to only have seven points is because most warship designs do not have a PMB. Therefore, the aft and forward PMB points are coincident with each other.

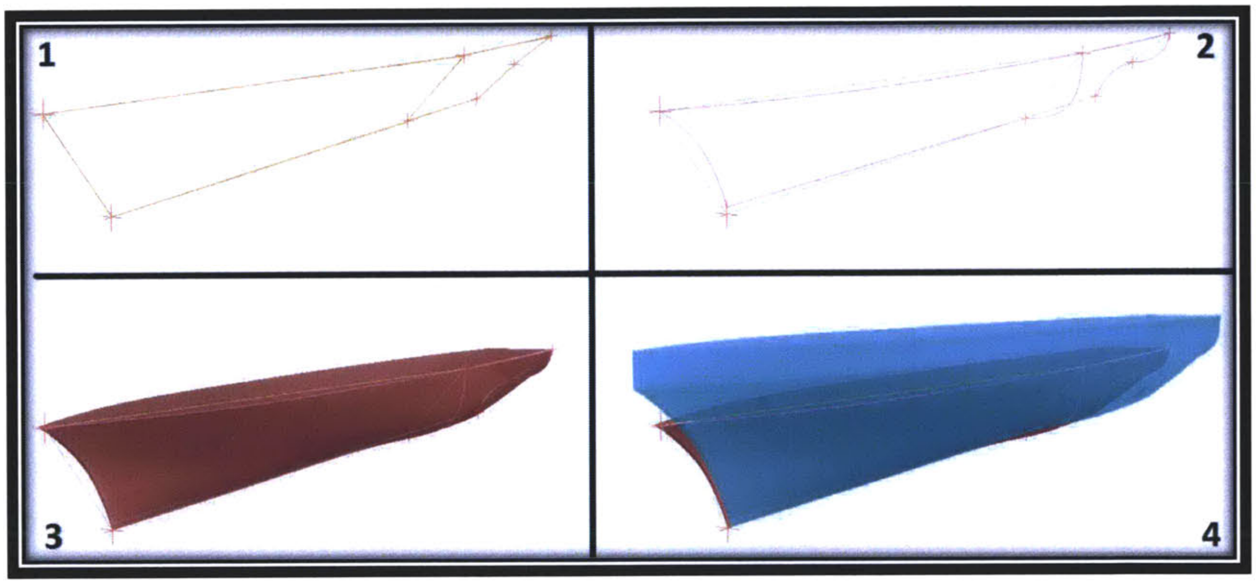
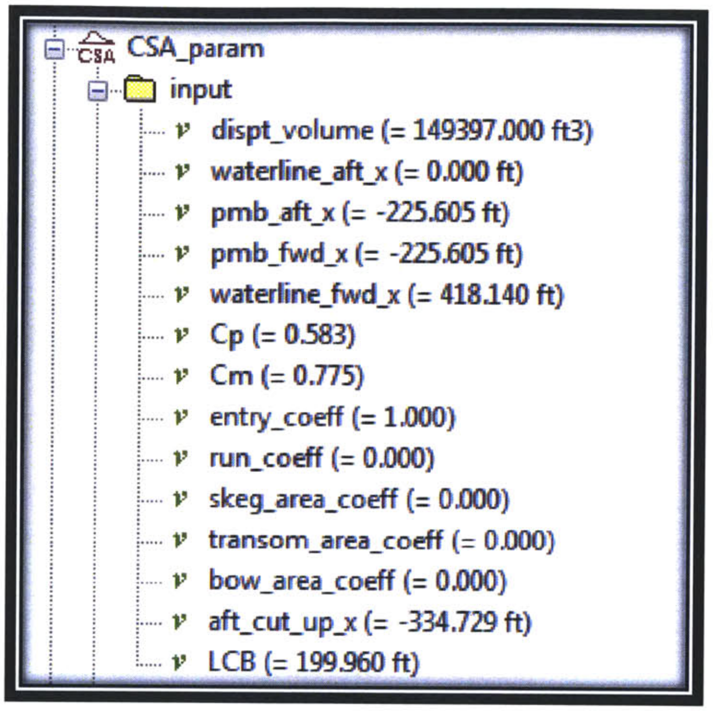


Figure 31: Illustration of Quick Hull Process

Next, as seen in step two of Figure 31, the QHM uses seven guide curves representing the transom, midship, bow, keel and deck edge aft of midship, and keel and deck edge forward of midship. Each curve is defined by nine points.

Paramarine then uses these control points and curves, in conjunction with the parameters displayed in Figure 32, to synthesize the parent hull as illustrated in step three of Figure 31.



```
CSA_param
  input
    dispt_volume (= 149397.000 ft3)
    waterline_aft_x (= 0.000 ft)
    pmb_aft_x (= -225.605 ft)
    pmb_fwd_x (= -225.605 ft)
    waterline_fwd_x (= 418.140 ft)
    Cp (= 0.583)
    Cm (= 0.775)
    entry_coeff (= 1.000)
    run_coeff (= 0.000)
    skeg_area_coeff (= 0.000)
    transom_area_coeff (= 0.000)
    bow_area_coeff (= 0.000)
    aft_cut_up_x (= -334.729 ft)
    LCB (= 199.960 ft)
```

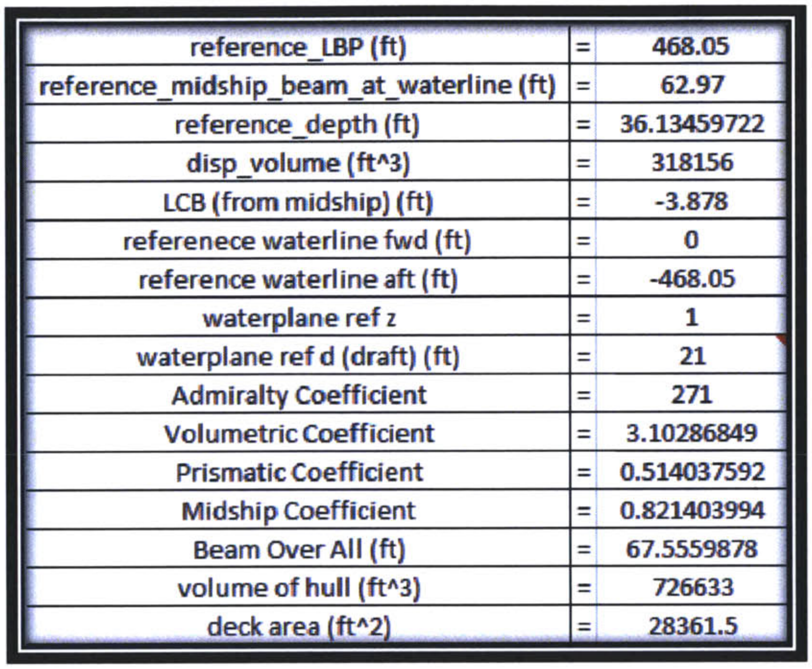
Figure 32: Parameters Required for QHM

Next the hull is uniformly scaled in length, beam, and depth in accordance with the calculations discussed in section 3.1.2 Calculations. Finally, if the required depth is greater than the depth of the newly scaled hull, a hull addendum is added atop the scaled hull. This is constructed automatically by the ESSDT by projecting the upper deck vertically by the necessary distance to overcome the shortfall. Once in the Paramarine model, this hull addendum may be altered to reflect more traditional hull flare or model the current trends toward tumblehome designs as the user prefers.

3.3.2 Hull Library

The key to being able to perform the QHM insertion is the hull library within the ESSDT. All available hulls are provided within the drop-down menu on the primary input page. Each hull within the library has its own, dedicated tab. On that tab are the special references for all control points and reference points for control curves of the parent hull. There is also the additional information required for the QHM as exemplified in Table 4, a selection from hull 5415, a DDG-51 style hull approved for public release.

Table 4: Example of Data in Hull Library from 5415 Hull



reference_LBP (ft)	=	468.05
reference_midship_beam_at_waterline (ft)	=	62.97
reference_depth (ft)	=	36.13459722
disp_volume (ft^3)	=	318156
LCB (from midship) (ft)	=	-3.878
referenece waterline fwd (ft)	=	0
reference waterline aft (ft)	=	-468.05
waterplane ref z	=	1
waterplane ref d (draft) (ft)	=	21
Admiralty Coefficient	=	271
Volumetric Coefficient	=	3.10286849
Prismatic Coefficient	=	0.514037592
Midship Coefficient	=	0.821403994
Beam Over All (ft)	=	67.5559878
volume of hull (ft^3)	=	726633
deck area (ft^2)	=	28361.5

Most of this information is required for the QHM, but some of it, such as the hull's volume and upper deck area, are included to assist with the bank calculations as previously discussed.

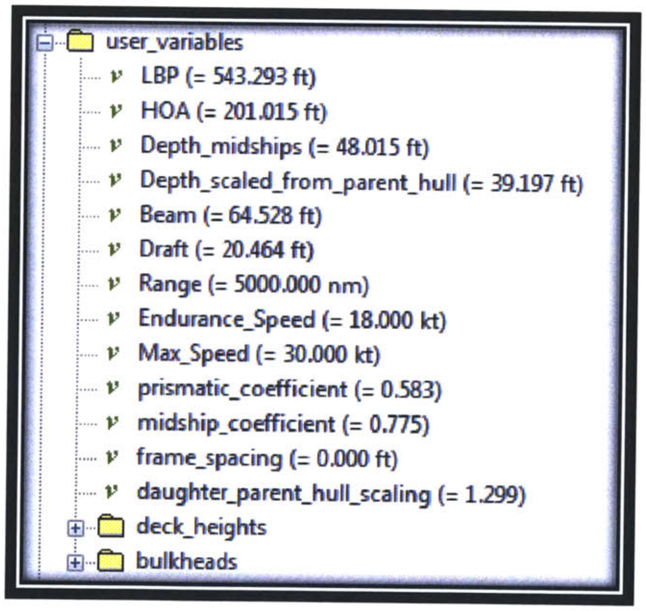
3.4 Outputs to Paramarine

Once the machinery plant, primary equipment, and hull type have been chosen, the deckhouse constructed, and the banks checked, the ESSDT is now ready to export its model to Paramarine. This involves two steps: inserting the requisite file structure (button number one in Figure 13: Final User Interface to Export Design to Paramarine) and actually constructing the model (button numbers two and three in Figure 13: Final User Interface to Export Design to Paramarine)

3.4.1 File Structure

The file structure created by the ESSDT closely models the structure provided in GRC's Surface Ship Early Stage Design tutorial (Pawling, 2009). The fully populated and expanded file structure can be seen in Appendix C: File Structure Created Within Paramarine by the ESSDT. There are two significant deviations from the structure utilized in the tutorial. The first is the inclusion of a "user_variables" folder. The second is the separation of the traditional "design" folder into two folders, a "design" folder and a "design_reference" folder.

The "user_variables" folder is a central control area where significant ship's information can be accessed and altered. All analysis and modeling dependent upon the ship's information included in the "user_variables" folder is linked to these inputs. An example is shown in Figure 33. This allows the user to rapidly alter key aspects of the model the ESSDT has created within Paramarine.



```
user_variables
    LBP (= 543.293 ft)
    HOA (= 201.015 ft)
    Depth_midships (= 48.015 ft)
    Depth_scaled_from_parent_hull (= 39.197 ft)
    Beam (= 64.528 ft)
    Draft (= 20.464 ft)
    Range (= 5000.000 nm)
    Endurance_Speed (= 18.000 kt)
    Max_Speed (= 30.000 kt)
    prismatic_coefficient (= 0.583)
    midship_coefficient (= 0.775)
    frame_spacing (= 0.000 ft)
    daughter_parent_hull_scaling (= 1.299)
    deck_heights
    bulkheads
```

Figure 33: "User_Variables" Folder

Like the "user_variables" folder, the "design_reference" folder was created to assist in dealing with modifying a model not originally created within Paramarine. Components which are part of the actual, physical design such as the hull, deckhouse, and equipment in location are placed within the "design" folder. Items which the design references, such as the QHM for the parent hull or the layout grid for decks and bulkheads, are stored in the "design_reference" folder. This allows for easier determination of what actually comprises the design and which components are there for model guidance.

3.4.2 Model Construction & Analysis

Once the file structure is in place, the ESSDT inserts the ship model into Paramarine. This model within Paramarine consists of references, the actual model, and analysis items. Most of the reference items are included in the previously discussed "user_variables" folder and "design_reference" folder. There is also an equipment library consisting of all major equipment placed upon the ship. It is located in the "setup" folder.

The physical model consists of the items within the "design" folder. This includes the hull and deckhouse as well as initial bulkheads and decks. The bulkhead and deck locations can be rapidly re-located within the sub-folders "deck_heights" and "bulkheads" in "user_variables". The model also contains all major equipment items located and modeled within the ship as input from the ESSDT.

For analysis, the initial Paramarine model has preliminary weight assessment, speed and powering, and trim and stability analysis. The speed and powering analysis is automated, and the initial model is populated with a power vs. speed curve for the design. Weight and trim and stability require a couple of links which are not automated. The details of these links and how to complete them are covered in Appendix A: Program Instruction Manual. Once completed, however, the model then contains a preliminary weight assessment and trim and stability analysis with a GZ-curve.

It is important to note that this is a preliminary analysis. For greater accuracy in design analysis, greater detail still needs to be developed within the Paramarine model. Of particular note is the lack of modeled tanks. Fuel tanks are assumed to have a mass centroid based upon the defaults within the ESSDT. However, at this point the model has all of the requisite structure for the user to begin to define actual tank locations and sizes within the Paramarine model. Additionally, there is no structural analysis, maneuvering analysis, or a floodable length curve produced. However, all of the pieces are in place. So a user familiar with Paramarine can continue with the model development and analysis from this point on.

(THIS PAGE INTENTIONALLY LEFT BLANK)

4.0 Case Studies and Tradeoffs

As a preliminary assessment of the ESSDT and its ability to be used in analysis, two case studies were run. The first case study was an attempt to model a ship similar to the DDG-51 Flight IIA. A starting model was produced by selecting a load-out and geometry similar to the DDG-51 Flight IIA, and the resulting model was compared to the current production ship. The second case study was designed to determine the ESSDT's ability to rapidly perform a trade-off comparison between two similarly equipped ships. Two medium sized surface combatants were constructed with differently sized engine plant arrangements. The resulting models were then compared.

4.1 Mimicking DDG-51

A potential application of the ESSDT is its ability to be used to quickly synthesize an initial stage design for further development. This could allow a naval engineer to leverage their time by producing a model which can be further designed in depth and analyzed in Paramarine in an expeditious manner. In an attempt to assess this application, the ESSDT was used to attempt a rapid synthesis of a ship similar to the DDG-51 Flight IIA. The resulting ship in Paramarine, shown in Figure 34, was then compared to the historically developed DDG-51 Flight IIA.

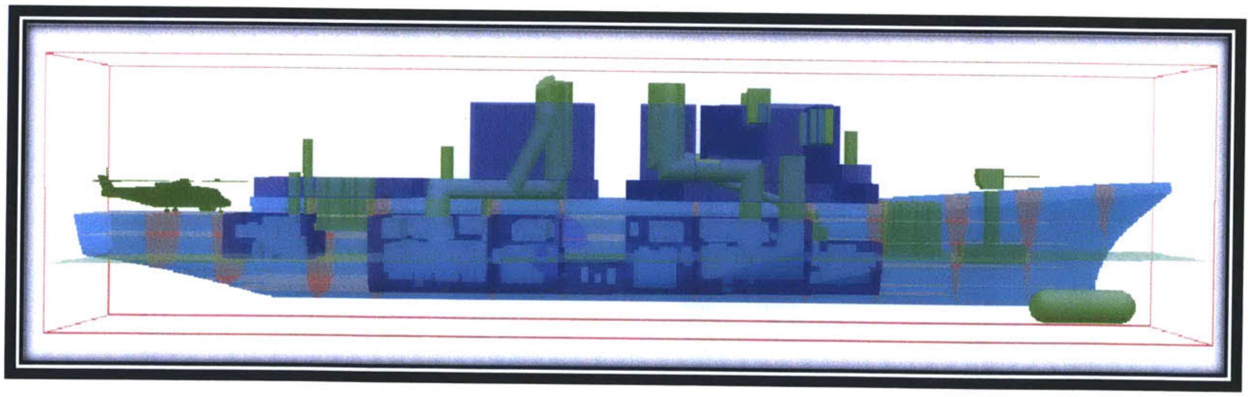


Figure 34: Paramarine Model Produced in Case Study 2

The inputs used to mimic a DDG-51 can be found in Appendix D: Input for Case Study 1, Mimicking DDG-51. Since the IPSDM was designed to produce only IPS plants, plant "IPS_MED_SURF_COMB_LONG" (Jurkiewicz, 2012) was chosen due to a similar size and overall geometry to the current flight of DDG-51s engine room arrangement. Figure 35 and Figure 36 display the resulting weight, power, and volume banks when the inputs for the DDG-51 were placed into the ESSDT.

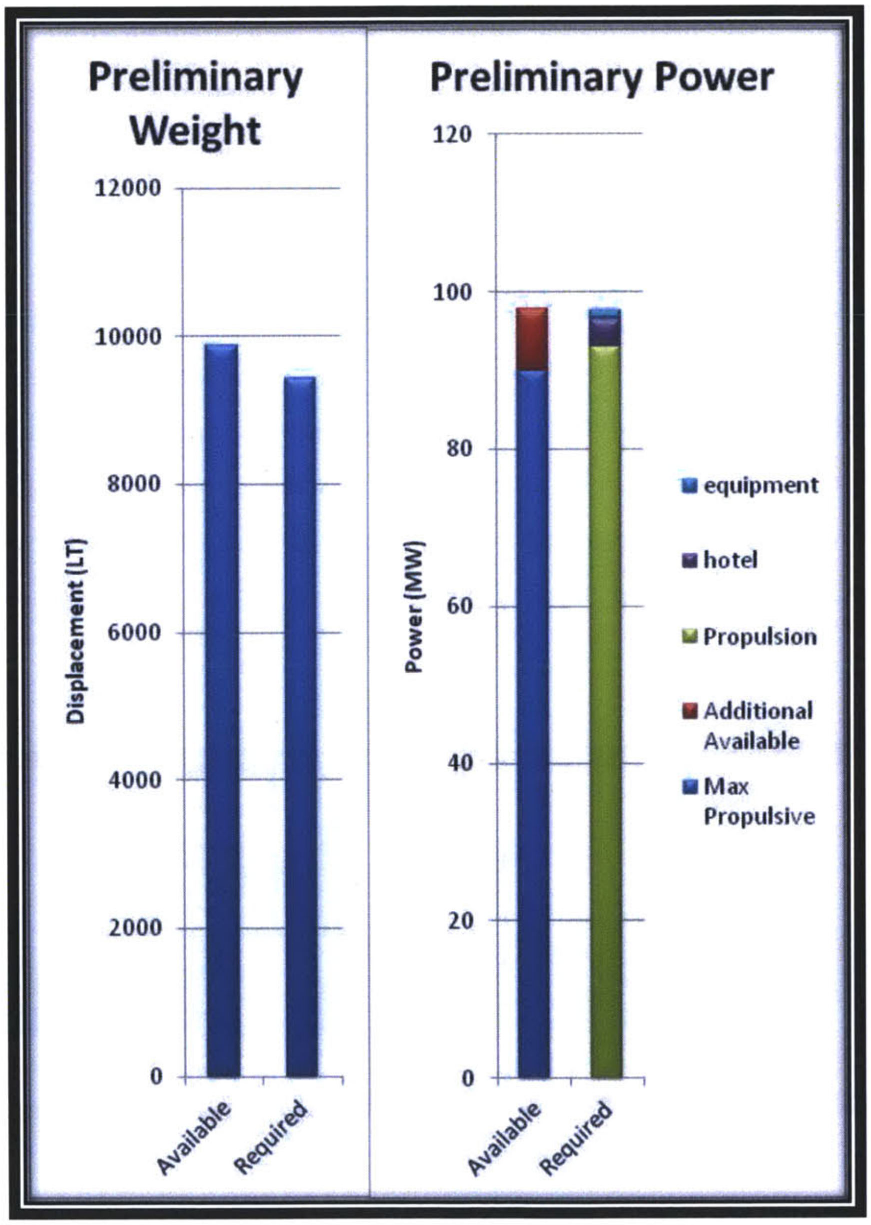


Figure 35: Weight & Power Bank for Case Study 2

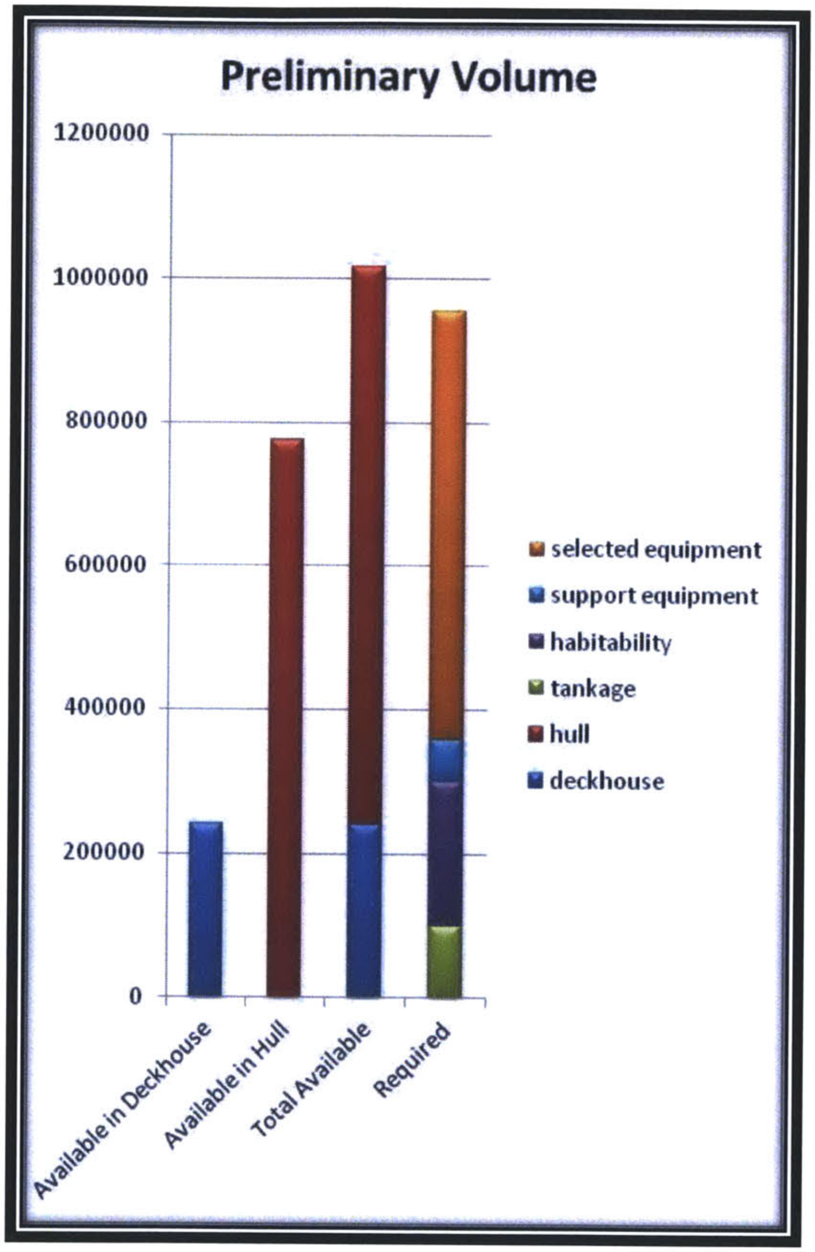


Figure 36: Volume Bank for Case Study 2

The General Dimensions of the final model in Paramarine can be seen in Table 5. It was found that the ESSDT produced a model with major dimensions within 5% of DDG-51 Flight IIA. Total time to produce this model was ~1 hour to construct the deckhouse and ~30 minutes for all other input. The resulting model did have a GM_t of only 4.2 ft (desired value is typically ~10% of beam) and a trim by the bow of approximately 0.69 degrees. However, it should be noted that these mildly undesirable stability values came from the analysis of a model with all tanks modeled with one centroid in default position. As a naval engineer moves forward with the design, this preliminary analysis shows that it would be desirable for tanks, when formally defined, to be placed overall lower and slightly to the aft of the current default position.

Table 5: Results of Case Study 1

	DDG-51 Flight IIA (Wikipedia, 2012)	ESSDT Produced Model	% Diff
Displacement	9,200 LT	9,120	-0.87%
Length Overall	509 ft	513 ft	0.79%
Beam	66 ft	69 ft	4.55%
Draft	30.5 ft	29.6 ft	-2.95%

Case study one successfully performed a preliminary modeling of a surface combatant similar to DDG-51 Flight IIA in a relatively short period of time. Although the resulting model needed further refinement, it demonstrated a feasible starting point for more in-depth model development and analysis.

4.2 Trade-off Comparison of Different Sized Power Plants

If a smaller power plant is placed within a ship, there are three likely effects on the final design, the ship as a whole will be able to be smaller, it will need less power, and it will probably be less expensive. Smaller ships require less power to reach the same speeds as larger ships. So, by reducing the size of the power plant, does the reduction in propulsive requirements compensate for the reduction in producible power? In other words, by reducing the power plant size, can a smaller, equally equipped ship go just as fast and have just as much power available for additional equipment as a larger ship with a larger power plant and larger price tag?

To address this issue, two similarly equipped medium-sized surface combatants were designed using the ESSDT. Each used the "frigate" hull from the ESSDT hull library and was equipped with the following items:

- 64 Cell VLS
- 128 Cell VLS
- 5" Deck Gun
- Two Phalanx Close-in-Weapons-Systems
- SQS 53-C Sonar
- Four SPY 1D Fixed Array Radars

Both ships were initially designed with the parameters in Table 6.

Table 6: Initial Specifications for the Two Medium-Sized Surface Combatants Analyzed

	LARGER SHIP	SMALLER SHIP
Max Speed	30 knots	30 knots
Range	5000 nm	5000 nm
Crew Size	340	340
Endurance	60 days	60 days
Cruising Speed	18 knots	18 knots
Installed Power	108.5 MW	103.7 MW
Initial Displacement	9460 LT	9387 LT

Figure 37 displays the layout of the two ships as it appeared in both the ESSDT and final modeling in Paramarine, and Figure 38 and Figure 39 display the resulting bank analysis from the synthesis of each ship.

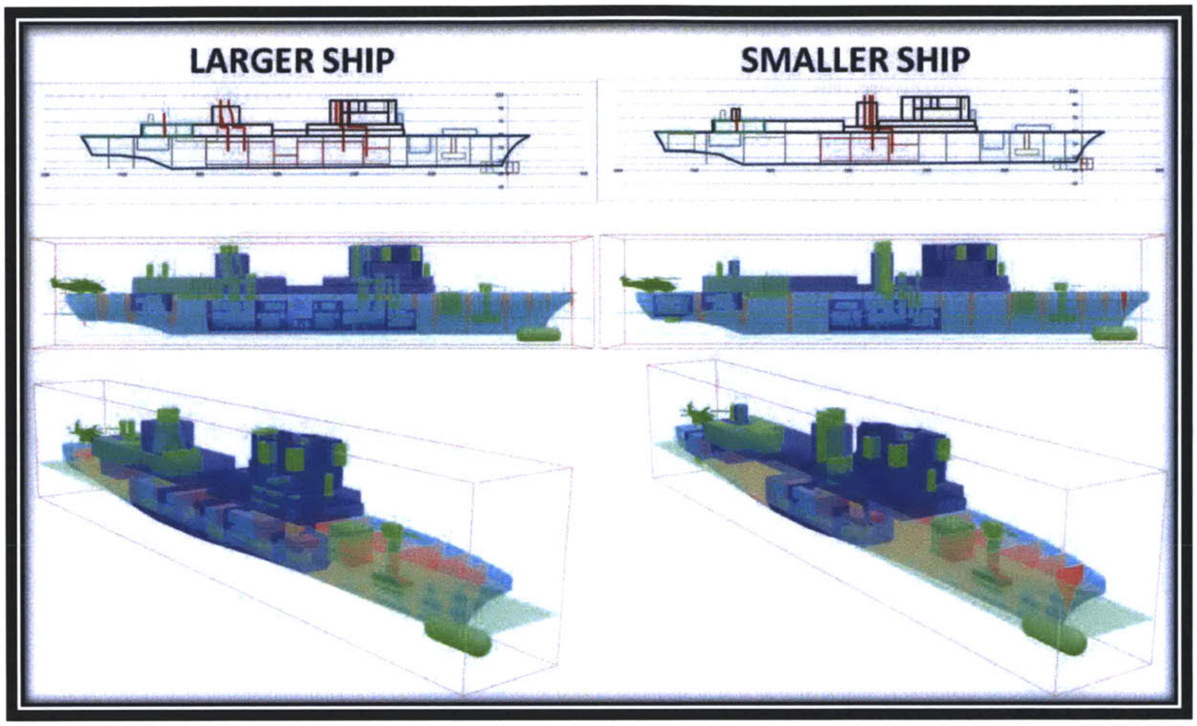


Figure 37: Layout of the Two Medium-Sized Surface Combatants Analyzed

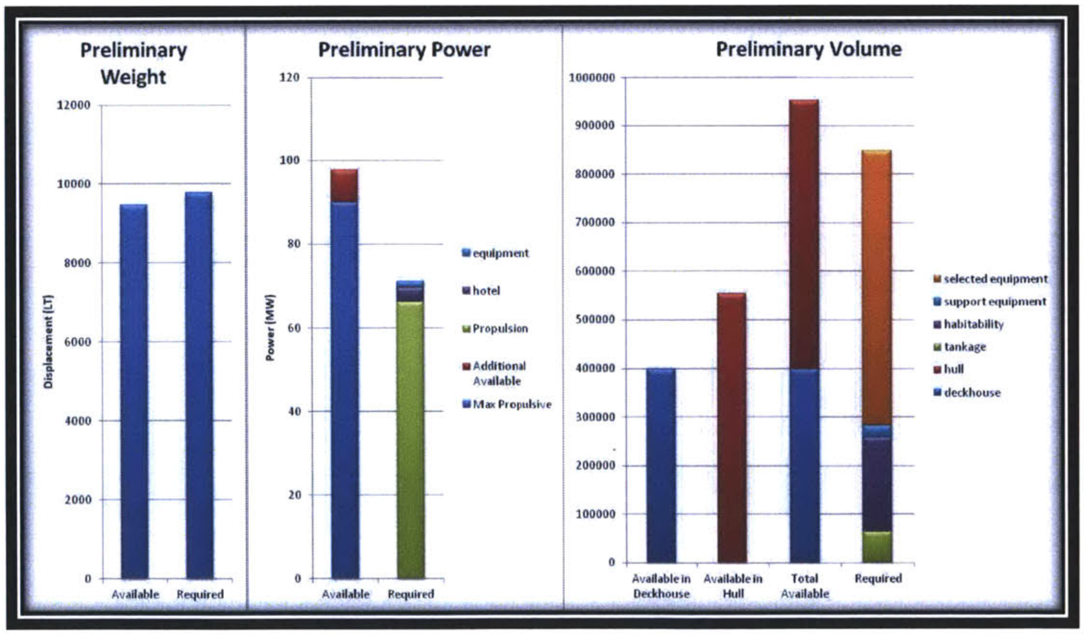


Figure 38: Bank Assessment, Larger Ship

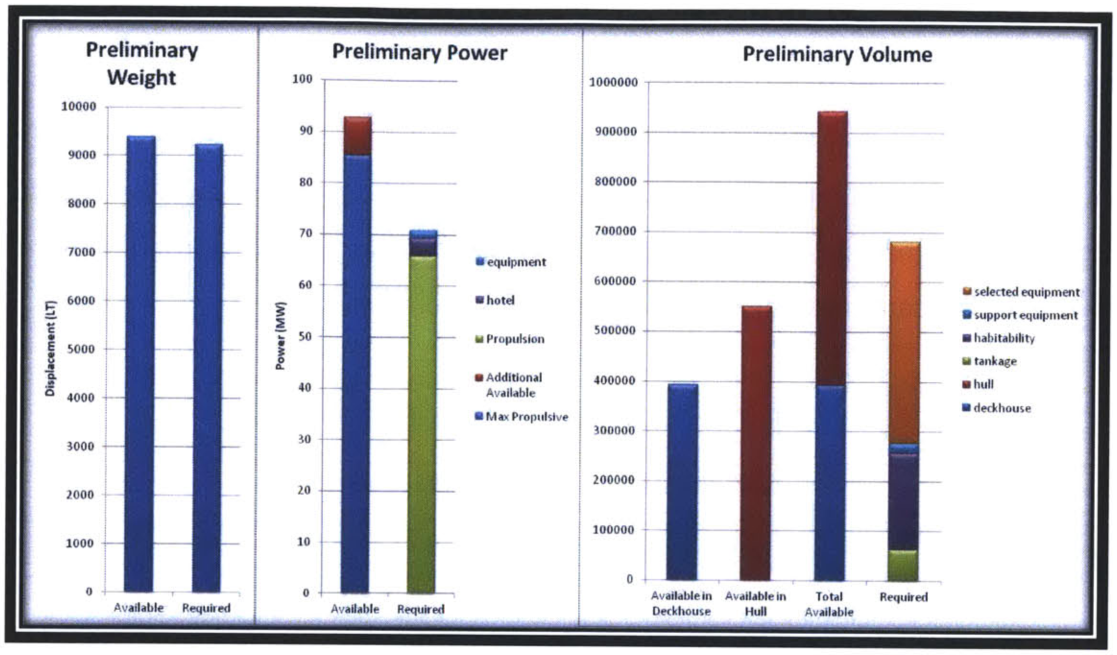


Figure 39: Bank Assessment, Smaller Ship

Figure 40 shows a comparison of the two ships' powering curves and the powering curve of the parent hull they were both based upon.

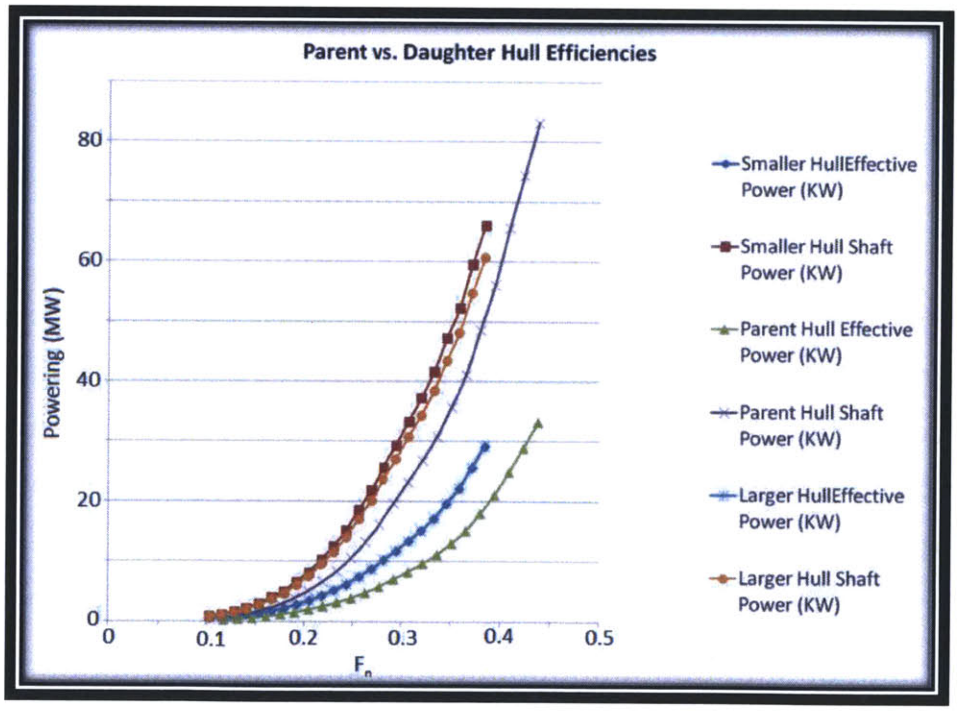


Figure 40: Comparison of Power Curves

Table 7 displays the measured outputs from each of these two ships. Based upon this analysis, this particular trade-off appears to not be profitable from a capability stand-point since the slight size difference garnered by a reduction in beam of 0.2 ft results in less power available for ship's systems. Of course this would have to be coupled with an estimated cost difference of the two plants to determine the value of the trade-off more definitively.

Table 7: Comparison of Case Study 2 Output

	LARGER SHIP	SMALLER SHIP	% DIFF
Trim	0.51°	0.76°	32.9%
GM_T	6.7 ft	6.7 ft	0%
Installed Power	108.5 MW	103.7 MW	4.4%
Initial Displacement	9460 LT	9387 LT	0.8%
Final Displacement	9449 LT	9376 LT	0.8%
LBP	544.7 ft	543.3 ft	0.3%
Beam	64.7 ft	64.5 ft	0.3%
Draft	20.5 ft	20.5 ft	0%
Additional Power Available	37.1 MW	26.7 MW	28%

While this particular analysis only showed minor differences, it should be noted that a previous version of the ESSDT which allowed independent scaling of the hull's dimensions showed much greater trade-offs between the two models. This was because the ability to scale length and beam independently allowed the models to capture the advantage of the reduction in stack length of the smaller plant.

This independent scaling was abandoned since it potentially led to inefficient hull designs by altering the form of the parent hull too greatly. However, this earlier analysis did demonstrated that if the hull library were more populated, and different shaped hulls were allowed to be selected to capture the reduction in stack length allowed with the smaller plant, this same analysis may lead to more significant results.

Total time to run this comparative analysis between these two ships was approximately 30 minutes. This demonstrated how the ESSDT could be used to rapidly model a design trade-space.

(THIS PAGE INTENTIONALLY LEFT BLANK)

5.0 Conclusions

5.1 General Conclusions

As shown by the case studies, the ESSDT is a potentially valuable design bridge between the earliest stages of design development and the ability to perform complex analysis of initial ship design ideas. It allows quick comparison between different design concepts and trade-offs. It also allows a designer to take a very primitive idea of what design options might be desired and place them into a tool like Paramarine such that all of the power of that tool is available for analysis.

The ESSDT accomplishes these tasks in such a way as to allow the incorporation of the flexibility afforded by the development of IPS. By moving away from the more traditional parametrics and towards inclusion of the actual selected equipment, the ESSDT enables a more reliable preliminary analysis than that afforded by a purely parametric analysis. Since there have not been enough IPS ships to form the basis of all of these parametrics as used in more traditional approaches to Naval Engineering, this approach may be a valuable tool for the foreseeable future.

5.2 Areas of Future Study & Development

In the process of developing and testing the ESSDT, it was determined that there are a number of areas where the concepts developed with the ESSDT would greatly benefit from further development and research. These include further development of the user interface, inclusion of zoning in the model construction, development of an electrical distribution module, expansion of existing libraries, and further development of the automated Paramarine analysis.

5.2.1 Refining Interface

In an attempt to use the ESSDT to rapidly develop ship models for analysis, one of the most inhibiting factors is the current interface. The requirement to place equipment and construct geometry in tabular form and continuously update the file to see the image of what's being produced is cumbersome at best. An example would be how defining the deckhouse geometry in case study one represented the bulk of the time used in that process. If the user interface were more graphics based, allowing users to click and drag equipment and geometry into place, the overall efficacy of the ESSDT could be greatly enhanced.

5.2.2 Inclusion of Zoning

The current manifestation of the ESSDT does not account for electrical, cooling, or fire zoning considerations within the design produced. This is an aspect of ship design which can be very helpful to the designer to define early on. Some consideration of zoning has been used in related programs such as the IPSDM (Jurkiewicz, 2012), and the CSDT (Fiedel, 2011). Developing a method to account for zoning within the ESSDT would be a valuable enhancement to the program. One way of accomplishing this would be to combine the ESSDT and CSDT such that a design created in the ESSDT could then be further developed by the CSDT.

5.2.3 Development of an Electrical Distribution Model

The CSDT provided a model of how the family of tools including ESSDT, IPSDM, and CSDT can be used to design and develop a ship's cooling system. At this time, there is no tool of this form for the similar development of a ship's electrical distribution system. If a tool similar to the CSDT could be developed and incorporated into the ESSDT, this could provide another valuable asset for ship design and analysis.

5.2.4 Expansion and Completion of Equipment Library

Currently, the biggest issue for the ability of the ESSDT to model modern designs accurately is the lack of depth in its equipment library. Exact equipment requirements for power, weight, cooling, area, etc. are difficult to obtain in a non-distribution limited manner. Additionally, the main focus of this thesis was developing the interaction between interface, library, and Paramarine. With that in place, further development should be dedicated to the expansion of the ESSDT's library in both the number of potential equipment instances available and in the data for that equipment. Ideally, it would be optimal to develop future iterations of the ESSDT such that they can access larger, out-of-program equipment libraries such as the one being developed by the Electric Ship Research and Design Consortium (ESRDC).

5.3.5 Expansion of Hull Library & Re-Working of Hull Insertion Techniques

During the process of developing the ESSDT, several techniques were examined for the development of a hull form for the modeled ship. A method allowing scaling of the vessel longitudinally, transversely, and vertically independent of one another would be ideal for placement and location analysis of the actual equipment. However, using a parent hull limits the independent scaling in order to avoid deviating too greatly from the hydrodynamic efficiency of the parent. The ESSDT could potentially be a more powerful tool if a method could be developed to individually scale the three ship's dimensions while obtaining an efficient hydrodynamic design for the hull. Barring this, a significant expansion of available parent hulls within the hull library would allow greater flexibility to the user and allow a greater range of viable analyses.

5.2.6 Development of Paramarine Analysis

In this version of the ESSDT, powering, weight accounting, and stability and trim analysis have been entirely or partially automated within Paramarine. However, Paramarine is a powerful tool capable of much greater depth of analysis. Due to time limitations, more of these analyses, such as floodable length, structural analysis, tank placement, etc. were not able to be automated. In the ESSDT's continued development, it would be highly recommended to continue to expand the amount of automated analysis it can perform in Paramarine.

(THIS PAGE INTENTIONALLY LEFT BLANK)

References

Adams, B. S. (2012, March 28). *Raye Jean Jordan Montague*. Retrieved April 3, 2012, from Encyclopedia of Arkansas History and Culture: http://encyclopediaofarkansas.net/encyclopedia/entry-detail.aspx?entryID=5565

Belda, J. (1973). The FORAN System. *Proceedings ICCAS.* Tokyo.

Bowman, F. L. (2000, Fall). An Integrated Electric Power System: the Next Step. *Undersea Warfare.*

Carlson, C. M. (1982). Navy Computer-Aided Ship Design. *Proceedings of the IREAPS Technical Symposium* (pp. 343-352). San Diego: Institute for Research and Engineering for Automation and Productivity in Shipbuilding.

D.W.Taylor Model Basin. (1966). Computer-Aided Ship Design and Construction Symposium. Washington, D.C.

Fiedel, E. R. (2011, June). Cooling System Early-Stage Design Tool for Naval Applications. Cambridge, Massachusetts, USA.

Green, J., Bullen, S., Bovey, R., & Alexander, M. (2007). *Excel 2007 VBA Programer's Reference.* Indianapolis: Wiley Publishing, Inc.

Hurst, R. (1973). BRITSHIPS-An Integrated Design and Production System. *Proceedings ICCAS.* Tokyo.

Jurkiewicz, D. J. (2012, May). Modular Machinery Arrangement and Its Impact in Early-Stage Naval Ship Design. Cambridge, MA.

MIT Sea Grant College Program. (2012, April 3). *Electric Ship History*. Retrieved from ESRDC: http://esrdc.mit.edu/history/

Nick, E., & Parsons, M. G. (2007). Fuzzy Optimal Arrangement of Spaces within a Zone-deck Region of a Ship. Ann Arbor, Michigan, USA.

Nowacki, H. (2009). Five Decades of Computer-Aided Ship Design. *Computer-Aided Design*, 956-969.

Papanikolaou, A. (2009). Holistic Ship Design Optimization. *Computer-Aided Design*, 1028-1044.

Pawling, R. (2009). *Surface Ship Early Stage Design Training Course for Warship Design.* Gosport: Graphics Research Corporation.

Society of Naval Architects and Marine Engineers. (2003). *Ship Design and Construction.* Jersey City: SNAME.

TG 22.3 Destroyer Escorts. (2012, April 3). Retrieved from USS Guadalcanal Task Group 22.3 Association: http://candotg.org/TGDestroyerEscorts.htm

Wikipedia. (2012, March 24). *Buckley Class Destroyer Escort*. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Buckley_class_destroyer_escort

Wikipedia. (2012, March 19). *Oliver Hazard Perry class frigate*. Retrieved April 3, 2012, from Wikipedia: http://en.wikipedia.org/wiki/Oliver_Hazard_Perry_class_frigate

Wikipedia. (2012, March 18). *USS Zumwalt (DDG-1000).* Retrieved from Wikipedia: http://en.wikipedia.org/wiki/DDG-1000

(THIS PAGE INTENTIONALLY LEFT BLANK)

Appendix A: Program Instruction Manual

Introduction:

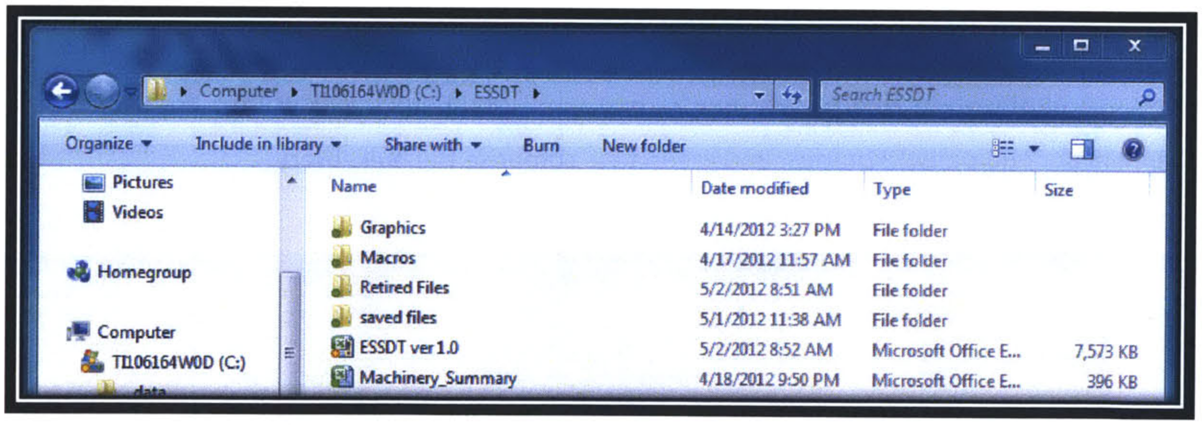
This manual provides an overview of how to use the Early Stage Ship Design Tool (ESSDT) Version 1.0 and what features are currently included. For more in-depth explanation of the underlying calculations and processes, please refer to the base document for which this manual is an appendix.

Requirements:

- Paramarine Version 7.0 or higher
- Microsoft Excel 2007 or later
- Integrated Power System Design Module (IPSDM)

Installation & Startup

Figure 1 illustrates the installed file structure for the ESSDT. Graphics for use with the ESSDT must be installed into their respective subfolder within the Graphics folder. For version 1.0, graphics to be inserted must consist of parasolids of a single solid body.

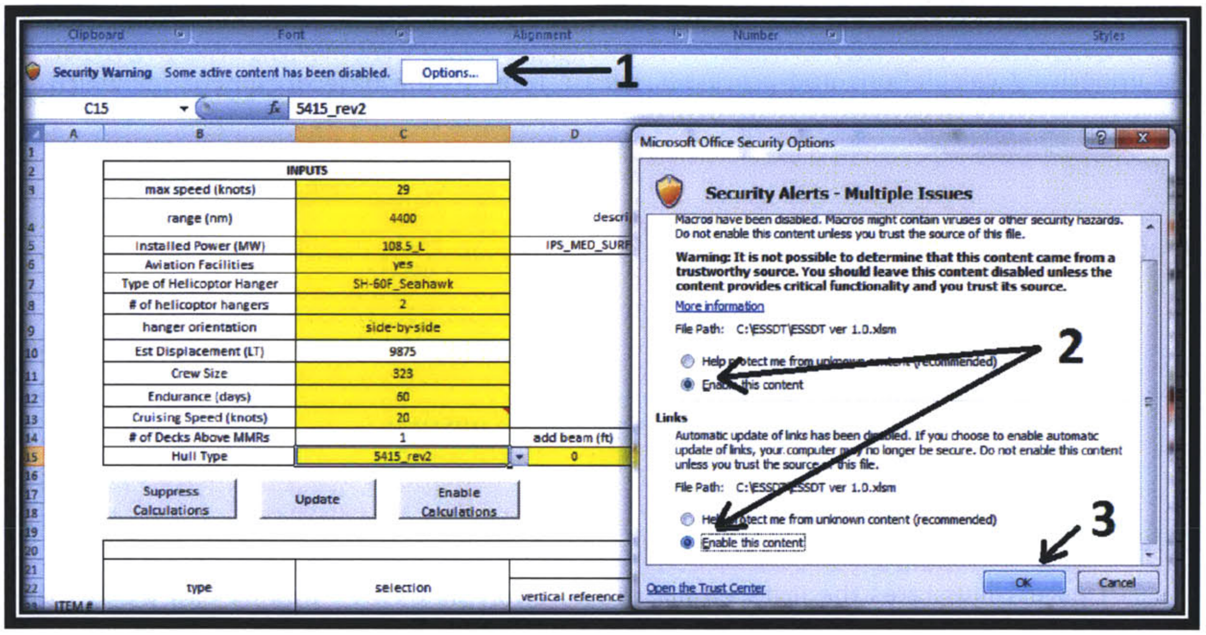


Name	Date modified	Type	Size
Graphics	4/14/2012 3:27 PM	File folder	
Macros	4/17/2012 11:57 AM	File folder	
Retired Files	5/2/2012 8:51 AM	File folder	
saved files	5/1/2012 11:38 AM	File folder	
ESSDT ver 1.0	5/2/2012 8:52 AM	Microsoft Office E...	7,573 KB
Machinery_Summary	4/18/2012 9:50 PM	Microsoft Office E...	396 KB

Figure 1: Installed File Structure

To use the ESSDT, begin by starting Microsoft Excel and opening the "Machinery_Summary" Excel file populated with the desired selection of power plant arrangements from the IPSDM (see IPSDM Manual for further detail). Then, open the main file

"ESSDT ver 1.0". Once open, enable macros and links, as shown in Figure 2, by selecting "options" (1), choosing to enable content (2), and selecting "OK" (3).



C15 5415_rev2

INPUTS	
max speed (knots)	29
range (nm)	4400
Installed Power (MW)	108.5_L
Aviation Facilities	yes
Type of Helicoptor Hanger	SH-60F_Seahawk
# of helicoptor hangers	2
hanger orientation	side-by-side
Est Displacement (LT)	9875
Crew Size	323
Endurance (days)	60
Cruising Speed (knots)	20
# of Decks Above MMRs	1
Hull Type	5415_rev2

add beam (ft) 0

Microsoft Office Security Options

Security Alerts - Multiple Issues

Macros have been disabled. Macros might contain viruses or other security hazards. Do not enable this content unless you trust the source of this file.

Warning: It is not possible to determine that this content came from a trustworthy source. You should leave this content disabled unless the content provides critical functionality and you trust its source.

More information

File Path: C:\ESSDT\ESSDT ver 1.0.xlsm

Help protect me from unknown content (recommended) [✓ Enable this content]

Links

Automatic update of links has been disabled. If you choose to enable automatic update of links, your computer may no longer be secure. Do not enable this content unless you trust the source of this file.

File Path: C:\ESSDT\ESSDT ver 1.0.xlsm

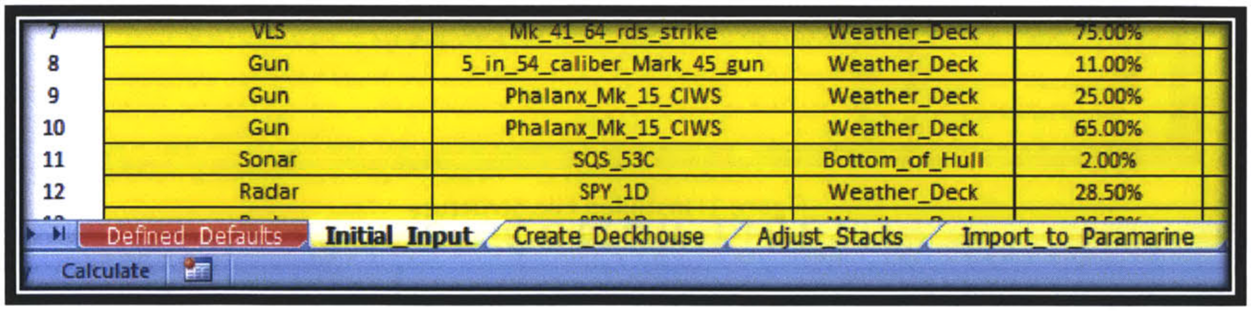
Help protect me from unknown content (recommended) [✓ Enable this content]

Open the Trust Center

Figure 2: Enabling Macros

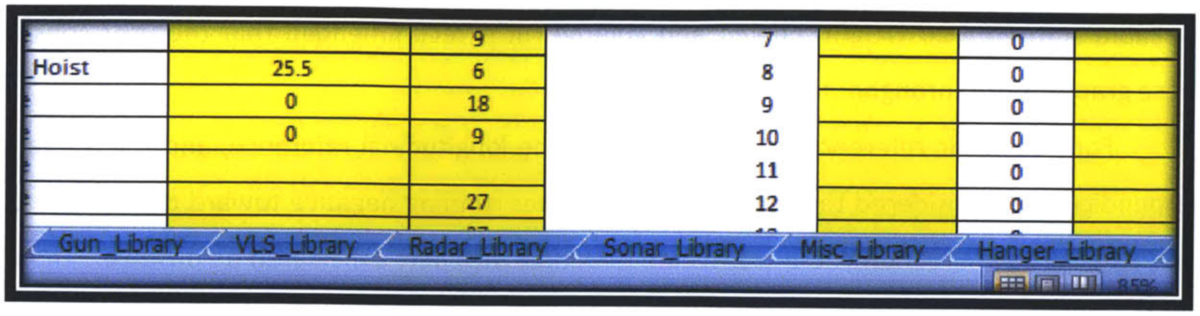
Input Overview:

The input for the ESSDT consists of a set of colored , selectable tabs at the bottom of the workbook as shown in Figures 3,4, and 5. Red tabs are defaults, yellow tabs are user inputs, blue tabs are equipment libraries, and green tabs comprise the hull library.



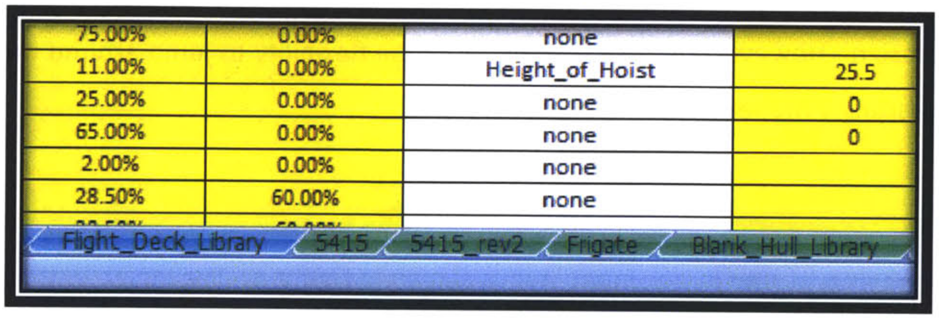
7	VLS	Mk_41_64_rds_strike	Weather_Deck	75.00%
8	Gun	5_in_54_caliber_Mark_45_gun	Weather_Deck	11.00%
9	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	25.00%
10	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	65.00%
11	Sonar	SQS_53C	Bottom_of_Hull	2.00%
12	Radar	SPY_1D	Weather_Deck	28.50%

Figure 3: Input Tabs



		9	7		0	
Hoist	25.5	6	8		0	
	0	18	9		0	
	0	9	10		0	
			11		0	
		27	12		0	

Figure 4: Input Tabs (cont.)



75.00%	0.00%	none	
11.00%	0.00%	Height_of_Hoist	25.5
25.00%	0.00%	none	0
65.00%	0.00%	none	0
2.00%	0.00%	none	
28.50%	60.00%	none	

Figure 5: Input Tabs (cont.)

Within each tab, input fields follow a similar color scheme. Cells highlighted in red are default values which may be changed should a more advanced or unusual design be desired. Cells highlighted in yellow are standard user input for ship model synthesis. Cells highlighted in blue are inputs for the equipment and hull libraries. These can be populated with the pertinent information for any equipment wished to be incorporated into the design or parent hulls for the ship.

Due to the large number of "lookup" functions within the program, changes, when input, may take several seconds to propagate through the program. To limit this delay, the ESSDT is set to suppress calculations. This means, as entries are made, the program must be updated for any changes to take effect. To accomplish this, the "calculate" button in the lower left corner of the program may be selected or any of the numerous "update" buttons throughout the program may be selected.

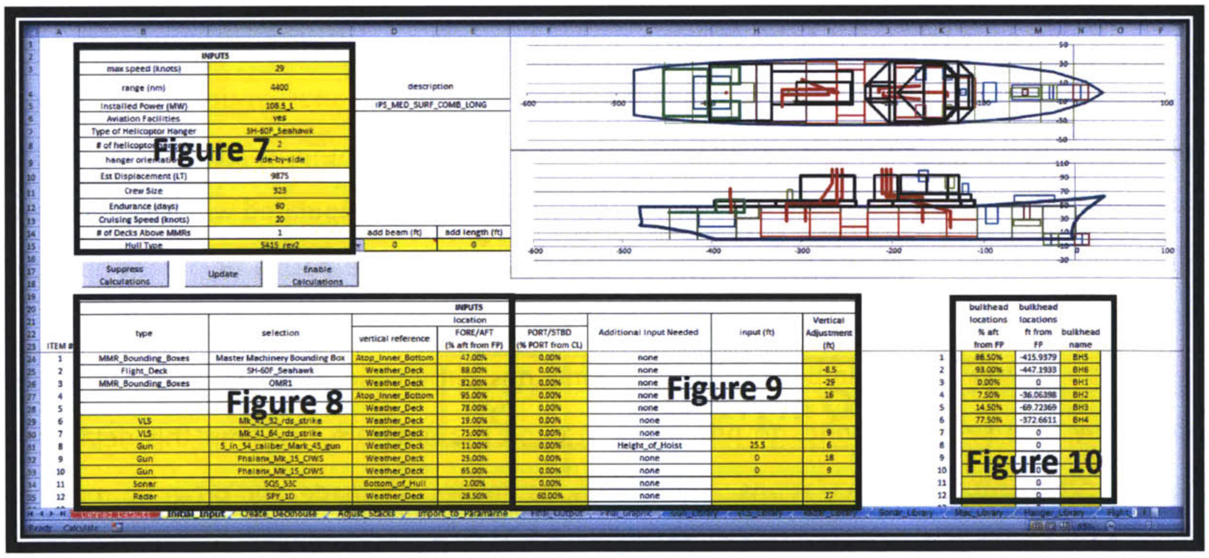
Although discussed in greater detail in the section concerning the "Import_to_Paramarine" tab, there are graphical representations of the designed ship's current available and required weight, power, and volume. It is recommended that the user consult these graphs often throughout the design process.

For geometric reference, the x-direction is the longitudinal reference, and the forward perpendicular is considered the zero point with values running negative toward the aft end of the vessel. The y-direction is the transverse direction with port considered as positive. The z-direction is the vertical direction measured as positive upward with the zero at the ship's baseline.

The following inputs are arranged in a logical order to proceed when developing an initial ship design. However, the ESSDT does allow the flexibility to jump around between tabs when synthesizing a model.

Initial Input:

An overview of the 'Initial_Input" tab is shown in Figure 6. The image in the upper-right corner is altered as the design is populated.



INPUTS	
max speed (knots)	29
range (nm)	4400
Installed Power (MW)	108.5_1
Aviation Facilities	yes
Type of Helicoptor Hanger	SH-60F_Seahawk
# of helicoptor bays	2
hanger orientation	side-by-side
Est Displacement (LT)	9875
Crew Size	323
Endurance (days)	60
Cruising Speed (knots)	20
# of Decks Above MMRs	1
Hull Type	5415_rev2

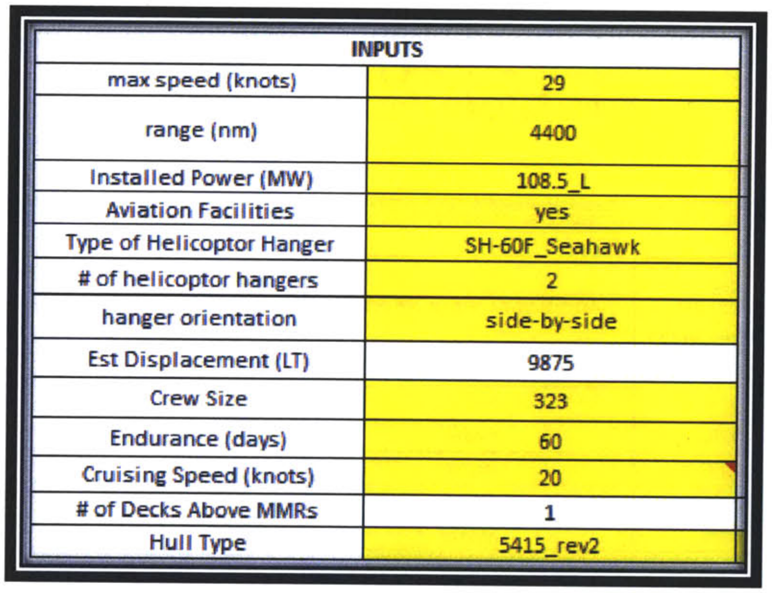
description	
IPS_MED_SURF_COMB_LONG	
add beam (ft)	add length (ft)
0	0

ITEM #	type	selection	INPUTS location vertical reference	INPUTS location FORE/AFT (% aft from FP)	PORT/STBD (% PORT from CL)	Additional Input Needed	input (ft)	Vertical Adjustment (ft)
1	MMR_Bounding_Boxes	Master Machinery Bounding Box	Atop_Inner_Bottom	47.00%	0.00%	none		
2	Flight_Deck	SH-60F_Seahawk	Weather_Deck	88.00%	0.00%	none		-8.5
3	MMR_Bounding_Boxes	OMR1	Weather_Deck	82.00%	0.00%	none		-29
4			Atop_Inner_Bottom	95.00%	0.00%	none		16
5			Weather_Deck	78.00%	0.00%	none		
6	VLS	Mk_41_32_rds_strike	Weather_Deck	19.00%	0.00%	none		
7	VLS	Mk_41_64_rds_strike	Weather_Deck	75.00%	0.00%	none		9
8	Gun	5_in_54_caliber_Mark_45_gun	Weather_Deck	11.00%	0.00%	Height_of_Hoist	25.5	6
9	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	25.00%	0.00%	none	0	18
10	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	65.00%	0.00%	none	0	9
11	Sonar	SQS_53C	Bottom_of_Hull	2.00%	0.00%	none		
12	Radar	SPY_1D	Weather_Deck	28.50%	60.00%	none		27

	bulkhead locations % aft from FP	bulkhead locations ft from FP	bulkhead name
1	86.50%	-415.9379	BH5
2	93.00%	-447.1933	BH6
3	0.00%	0	BH1
4	7.50%	-36.06398	BH2
5	14.50%	-69.72369	BH3
6	77.50%	-372.6611	BH4
7		0	
8		0	
9		[illegible]	
10		0	
11		0	
12		0	

Figure 6: Overview of Initial Input Page

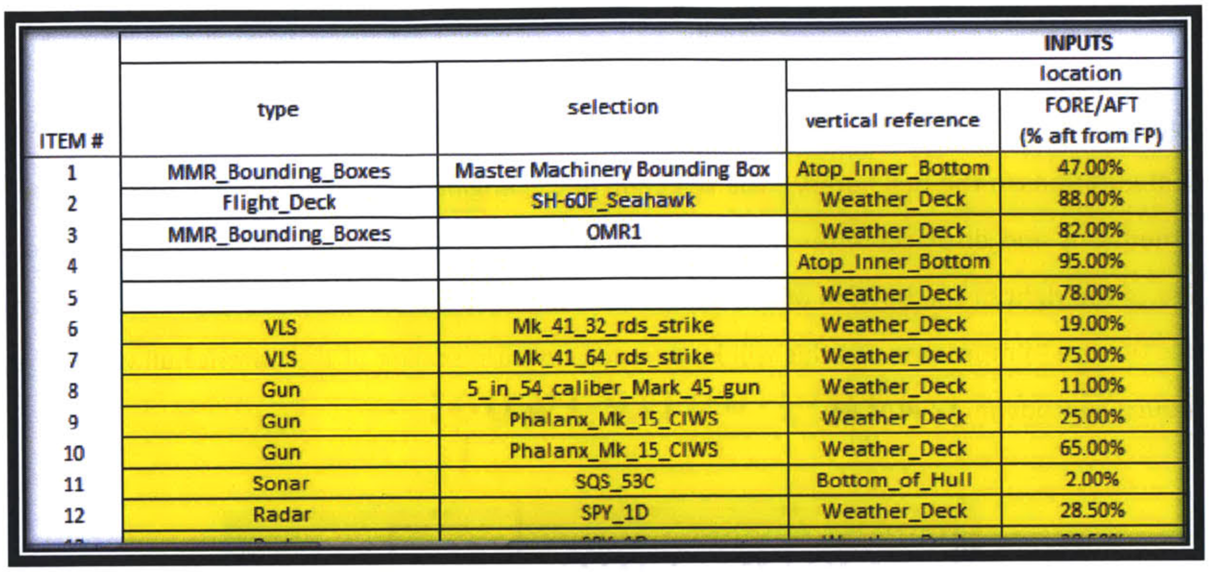
Figure 7 displays the initial input data for the ship. The selection for installed power provides a drop-down menu populated by all selectable plant configurations from the IPSDM output. Hull type similarly provides a drop-down menu populated by available hull configurations in the hull library. The selection of the engine plant and hull type sets the ship's dimensions and displacement. However, additional beam or draft may be added in the cells just to the right of the bottom of the initial inputs table. It is recommended to avoid the use of this feature in most cases as this will lead to non-uniform scaling of the parent hull which may reduce hydrodynamic efficacy.



INPUTS	
max speed (knots)	29
range (nm)	4400
Installed Power (MW)	108.5_L
Aviation Facilities	yes
Type of Helicoptor Hanger	SH-60F_Seahawk
# of helicoptor hangers	2
hanger orientation	side-by-side
Est Displacement (LT)	9875
Crew Size	323
Endurance (days)	60
Cruising Speed (knots)	20
# of Decks Above MMRs	1
Hull Type	5415_rev2

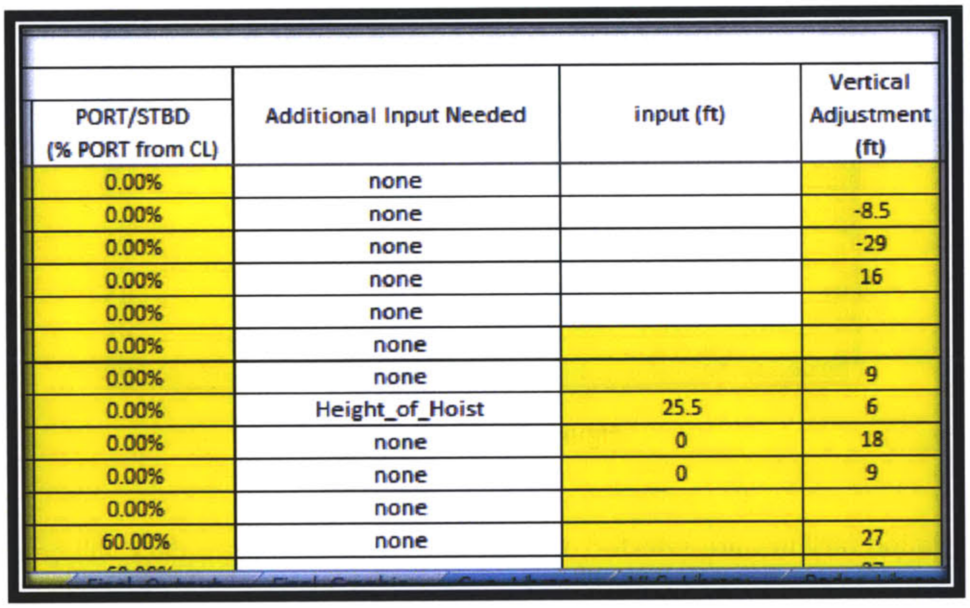
Figure 7: Initial Inputs

If aviation facilities are selected, the ability to position a flight deck will be enabled as seen in Figure 8 item #2 and in Figure 9 in the second row. In the initial inputs table, the user has the choice of 0-2 hangars, what type of helicopter they are sized for, and the orientation (side-by-side or fore-and-aft) if two are selected. The actual location and spacing between hangars is directed in the "Create_Deckhouse" tab discussed later.



ITEM #				INPUTS
			location	
	type	selection	vertical reference	FORE/AFT (% aft from FP)
1	MMR_Bounding_Boxes	Master Machinery Bounding Box	Atop_Inner_Bottom	47.00%
2	Flight_Deck	SH-60F_Seahawk	Weather_Deck	88.00%
3	MMR_Bounding_Boxes	OMR1	Weather_Deck	82.00%
4			Atop_Inner_Bottom	95.00%
5			Weather_Deck	78.00%
6	VLS	Mk_41_32_rds_strike	Weather_Deck	19.00%
7	VLS	Mk_41_64_rds_strike	Weather_Deck	75.00%
8	Gun	5_in_54_caliber_Mark_45_gun	Weather_Deck	11.00%
9	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	25.00%
10	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	65.00%
11	Sonar	SQS_53C	Bottom_of_Hull	2.00%
12	Radar	SPY_1D	Weather_Deck	28.50%

Figure 8: Equipment Selection, Left Side



PORT/STBD (% PORT from CL)	Additional Input Needed	input (ft)	Vertical Adjustment (ft)
0.00%	none		
0.00%	none		-8.5
0.00%	none		-29
0.00%	none		16
0.00%	none		
0.00%	none		
0.00%	none		9
0.00%	Height_of_Hoist	25.5	6
0.00%	none	0	18
0.00%	none	0	9
0.00%	none		
60.00%	none		27

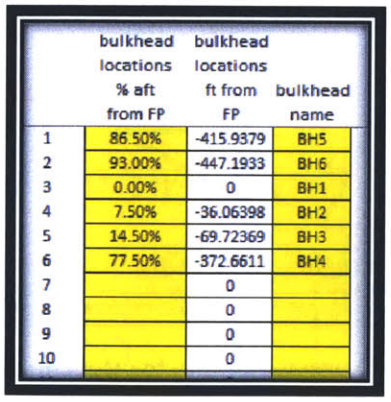
Figure 9: Equipment Selection, Right Side

The first line of the Equipment Selection Table (EST) is populated by the "MMR_Bounding_Box". This represents the Main Machinery Rooms (MMRs) and Auxiliary Machinery Rooms (AMRs) from the selected plant. They are moved together as a unit when placing them with the ship. Items 3-5 are populated with any Other Machinery Rooms (OMRs) which are part of the selected plant. These are placed independently of the MMRs, AMRs, and other OMRs. If aviation facilities were selected, then the EST allows the selection of a flight deck size based on information from the equipment library in row 2.

Additional selected equipment (item 6 on) is chosen by first selecting "type" followed by "selection". Both of these selections are populated from the available equipment within the equipment library. Locating the equipment consists of designating longitudinal, transverse, and vertical position. Longitudinal and transverse locations are selected as a percentage of the ship's dimension in that direction from either the forward perpendicular or centerline respectively. Vertical positioning is determined by selecting the bottom of the hull, atop the inner bottom, or the weather deck. An off-set from this selected vertical position may then be specified in the final column of the EST.

Some items may require an additional input as can be seen in Figure 9. In this example, it is the height of the hoist for a deck gun. This prompt appears when an equipment item requiring some further defining attribute is selected. If left blank, the value is assumed to be zero.

Figure 10 is the input for selecting bulkhead locations. The ship is automatically populated with the required bulkheads from the selected engine plant. Additional bulkheads are placed in this table as a percentage of ship's length from the forward perpendicular. In order for the ESSDT to properly recognize which bulkheads to insert, the user must list a name for each bulkhead to be incorporated in the design. This included the MMR bulkheads listed at the bottom of this bulkhead input table.

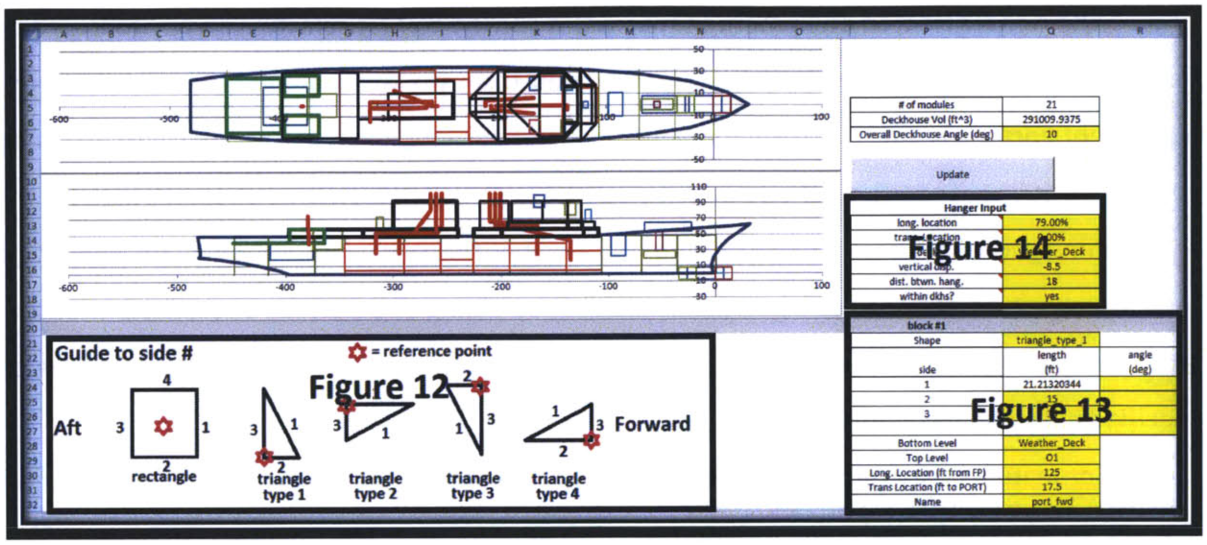


	bulkhead locations % aft from FP	bulkhead locations ft from FP	bulkhead name
1	86.50%	-415.9379	BH5
2	93.00%	-447.1933	BH6
3	0.00%	0	BH1
4	7.50%	-36.06398	BH2
5	14.50%	-69.72369	BH3
6	77.50%	-372.6611	BH4
7		0	
8		0	
9		0	
10		0	

Figure 10: Input for Bulkhead Locations

Create Deckhouse:

The initial input for creating the deckhouse is displayed in Figure 11.



# of modules	21
Deckhouse Vol (ft^3)	291009.9375
Overall Deckhouse Angle (deg)	10

Hanger Input	
long. location	79.00%
trans. location	0.00%
deck	Weather_Deck
vertical disp.	-8.5
dist. btwn. hang.	18
within dkhs?	yes

block #1		
Shape	triangle_type_1	
side	length (ft)	angle (deg)
1	21.21320344	
2	15	
3		
Bottom Level	Weather_Deck	
Top Level	01	
Long. Location (ft from FP)	125	
Trans Location (ft to PORT)	17.5	
Name	port_fwd	

Figure 11: Input for Creating the Deckhouse

The deckhouse is constructed of a series of up to 24 blocks constructed as either rectangles or triangles. Figure 12 displays the guidance for selecting, sizing, and locating the different shapes to construct the deckhouse. The reference point is the part of the shape placed by the longitudinal, transverse, and vertical placement as demonstrated in the Module Construction Table (MCT) in Figure 13.

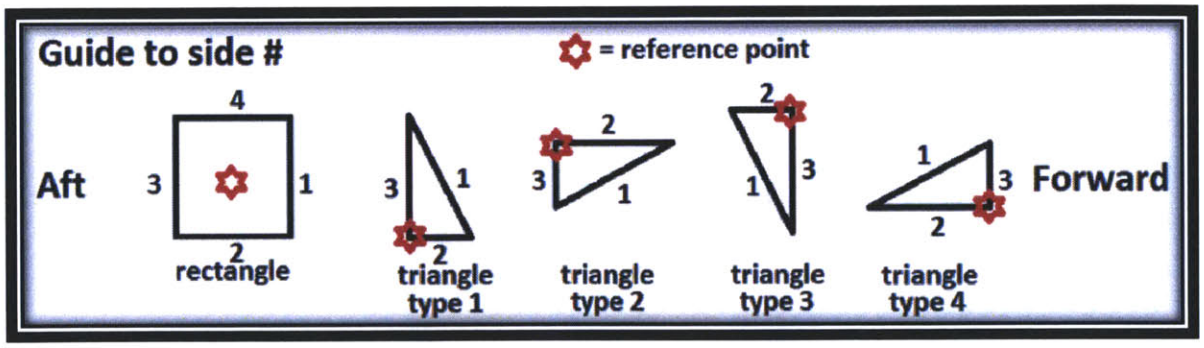
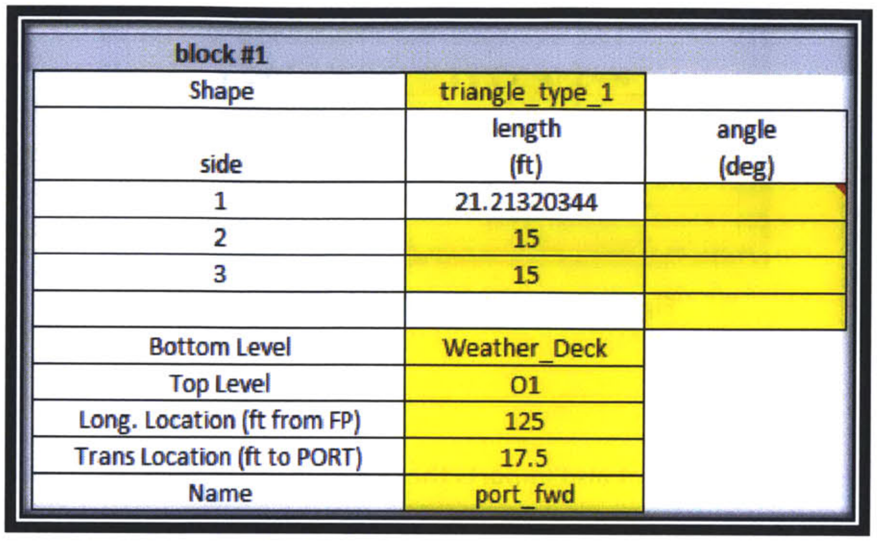


Figure 12: Shape Guidance for Deckhouse Creation

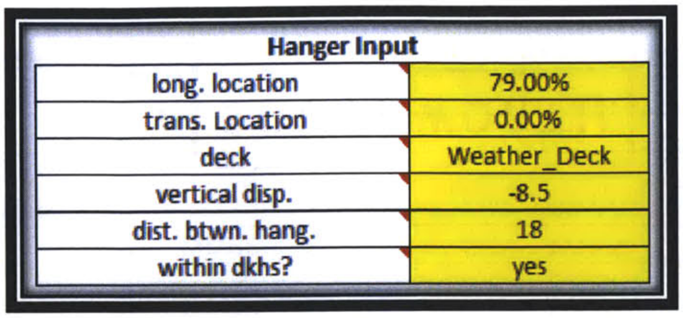


block #1		
Shape	triangle_type_1	
side	length (ft)	angle (deg)
1	21.21320344	
2	15	
3	15	
Bottom Level	Weather_Deck	
Top Level	01	
Long. Location (ft from FP)	125	
Trans Location (ft to PORT)	17.5	
Name	port_fwd	

Figure 13: MCT

When creating a block to be part of the deckhouse, first the shape must be selected from the five options displayed in Figure 12. Then, the longitudinal and transverse dimensions can be entered (sides 2 and 3 respectively). The vertical dimension is determined by selecting the bottom and top levels for the particular module. It can then be placed longitudinally and transversely within the ship. In sizing and placing, it is important to avoid any significant overlap between modules as this will result in "double counting" the available volume within the deckhouse. In order to be properly incorporated into the Paramarine model, the module must be given a unique name with no spaces or non-standard symbols. The angle input for each side is not, as of version 1.0, incorporated into the Paramarine model. At this time, input into this field has no effect.

The hangar input is displayed in Figure 14. The first four rows have the same effect for placement as their corresponding indications in the EST. The distance between the hangars gives the absolute difference between the hangars regardless of orientation. Selecting "within dkhs" is purely for accounting purposes. If "yes" is selected, the hangar volume will be subtracted from the available deckhouse volume.

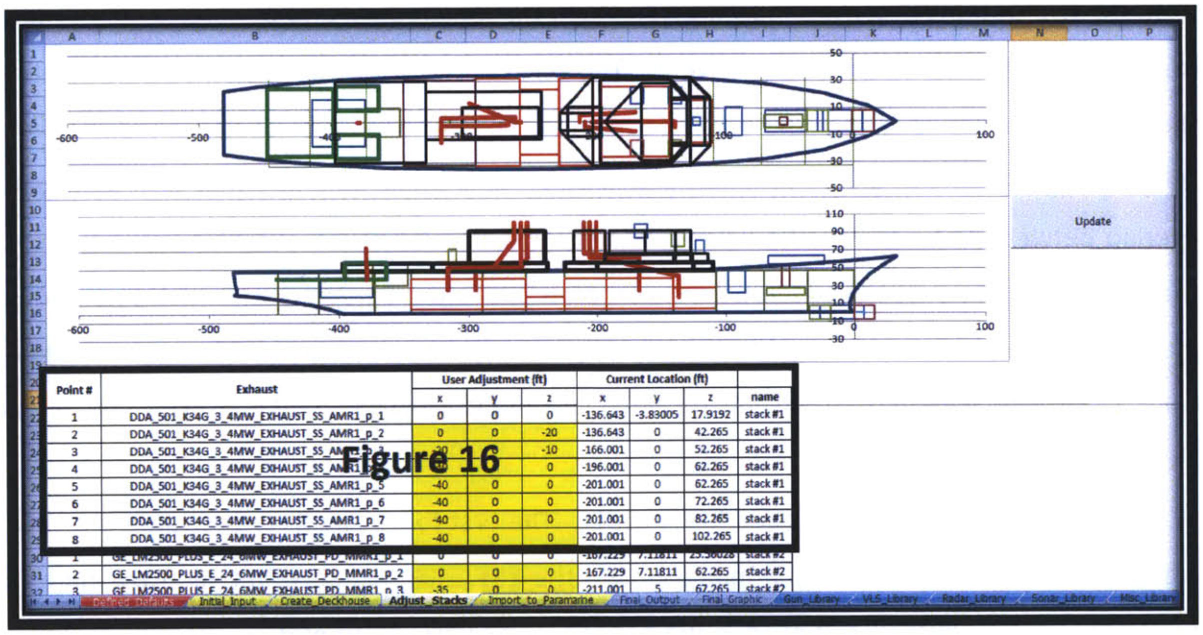


Hanger Input	
long. location	79.00%
trans. Location	0.00%
deck	Weather_Deck
vertical disp.	-8.5
dist. btwn. hang.	18
within dkhs?	yes

Figure 14: Input for Hangar Placement

Adjust Stacks:

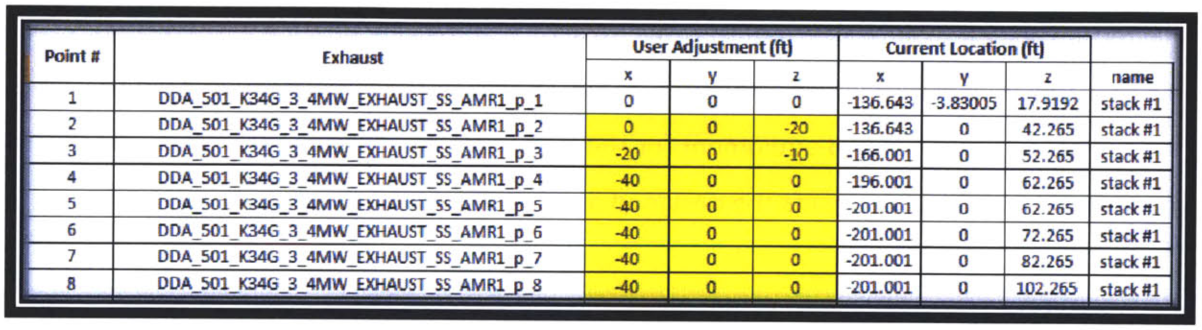
The selected engine plant also imports the requisite exhaust stacks. These are displayed as thick red lines on the ship displays in the ESSDT. The "Adjust_Stacks" tab allows the alteration of these imported exhaust stack paths. Figure 15 displays the "Adjust_Stacks" interface.



Point #	Exhaust	User Adjustment (ft)			Current Location (ft)			
		x	y	z	x	y	z	name
1	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_1	0	0	0	-136.643	-3.83005	17.9192	stack #1
2	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_2	0	0	-20	-136.643	0	42.265	stack #1
3	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_3	-30	0	-10	-166.001	0	52.265	stack #1
4	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_4	[illegible]	[illegible]	0	-196.001	0	62.265	stack #1
5	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_5	-40	0	0	-201.001	0	62.265	stack #1
6	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_6	-40	0	0	-201.001	0	72.265	stack #1
7	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_7	-40	0	0	-201.001	0	82.265	stack #1
8	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_8	-40	0	0	-201.001	0	102.265	stack #1
1	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_1	0	0	0	-167.229	7.11811	25.36028	stack #2
2	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_2	0	0	0	-167.229	7.11811	62.265	stack #2
3	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_3	-35	0	0	-211.001	5	67.265	stack #2

Figure 15: Input Tab for Adjusting Exhaust Stack Path

Figure 16 is a closer look at the input for adjusting the stack path. The original stack path consists of up to 8 control points defining the path for the exhaust stack. The user adjustment represents how far each of these eight control points deviates from their original point from the IPSDM output. For example, point number three in Figure 16 has been moved 20 feet aft and 10 feet down from its original position. To adjust the stacks, simply enter the corrections for each of the control points for each of the stacks until the paths are as desired. It is often helpful to have the deckhouse completed prior to this step.



Point #	Exhaust	User Adjustment (ft)			Current Location (ft)			
		x	y	z	x	y	z	name
1	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_1	0	0	0	-136.643	-3.83005	17.9192	stack #1
2	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_2	0	0	-20	-136.643	0	42.265	stack #1
3	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_3	-20	0	-10	-166.001	0	52.265	stack #1
4	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_4	-40	0	0	-196.001	0	62.265	stack #1
5	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_5	-40	0	0	-201.001	0	62.265	stack #1
6	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_6	-40	0	0	-201.001	0	72.265	stack #1
7	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_7	-40	0	0	-201.001	0	82.265	stack #1
8	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_8	-40	0	0	-201.001	0	102.265	stack #1

Figure 16: Table for Adjusting Stack Paths

Import to Paramarine:

Figure 17 is the output screen for export of the final model to Paramarine.

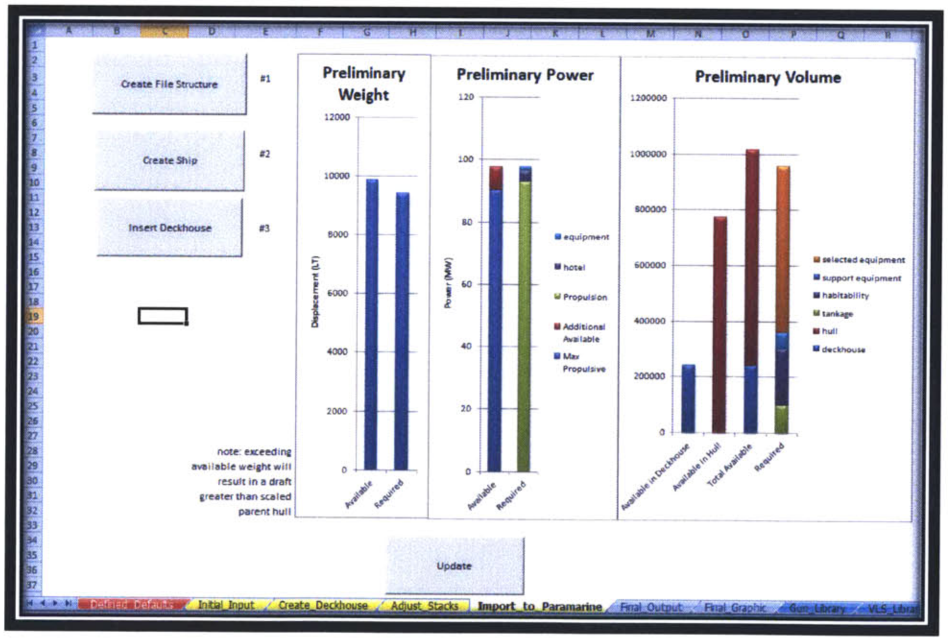


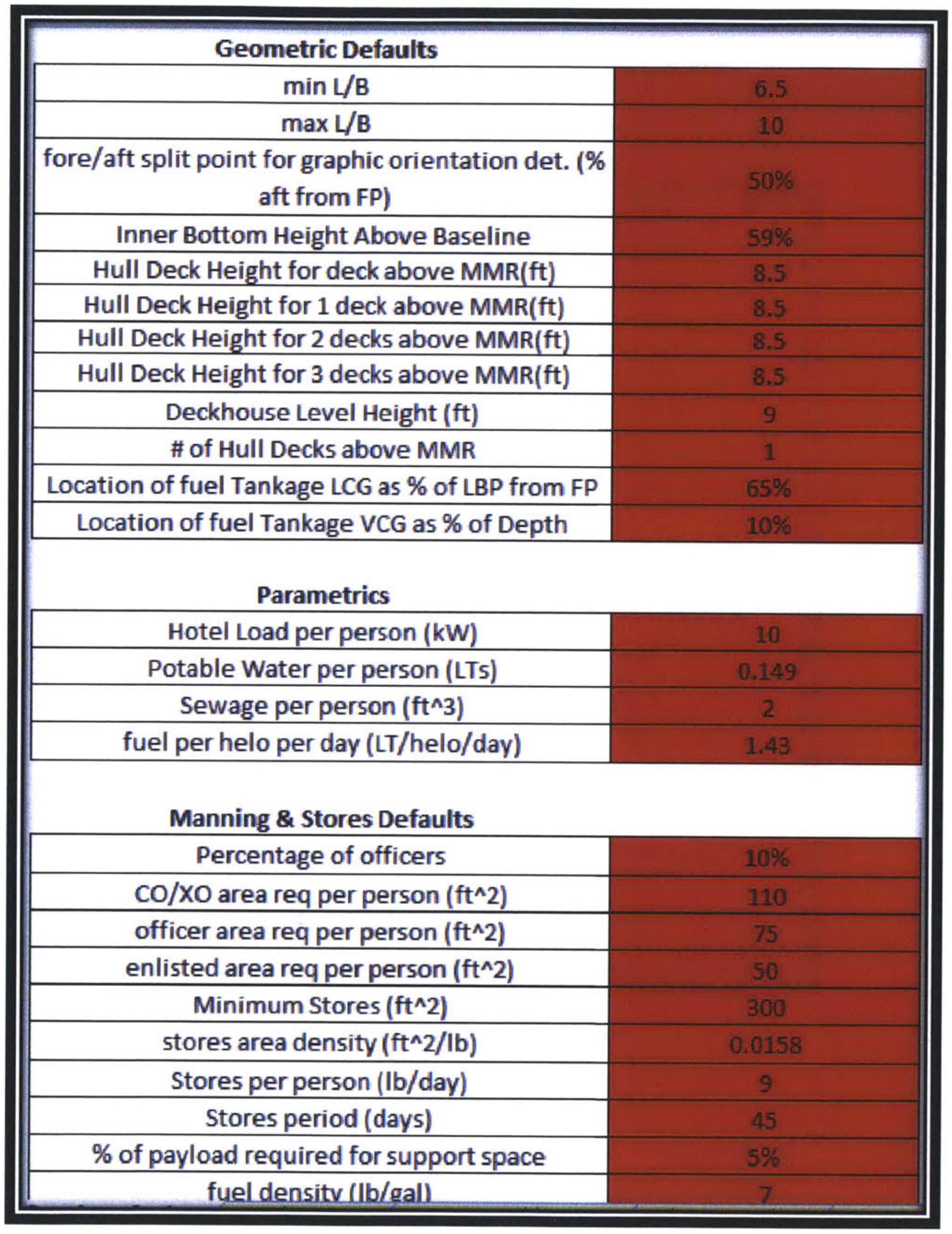
Figure 17: Import to Paramarine Tab

Each of the bar graphs displayed in Figure 17 is a representation of the balance between available and required values for the respective ship's attributes. These are rough estimates and have default margins built into them. Final analysis is left to Paramarine.

To create a ship in Paramarine, four steps must be accomplished. First, Paramarine 7.0 or later must be opened. Next, the first button in Figure 17, "Create File Structure", must be selected. This will create the requisite file structure within Paramarine. Then, the "Create Ship" button must be selected. This will populate the aforementioned file structure within Paramarine with the hull and equipment for the model designed within the ESSDT. Finally, the "Insert Deckhouse" button insets the designed deckhouse into the Paramarine model.

Defined Defaults:

The ESSDT also contains default values which can be altered by the user. These are located under the "Defined_Defaults" tab and can be seen in Figures 18-21. Changing these defaults is not recommended unless the user has a fairly intimate understanding of naval engineering.



Geometric Defaults

min L/B	6.5
max L/B	10
fore/aft split point for graphic orientation det. (% aft from FP)	50%
Inner Bottom Height Above Baseline	59%
Hull Deck Height for deck above MMR(ft)	8.5
Hull Deck Height for 1 deck above MMR(ft)	8.5
Hull Deck Height for 2 decks above MMR(ft)	8.5
Hull Deck Height for 3 decks above MMR(ft)	8.5
Deckhouse Level Height (ft)	9
# of Hull Decks above MMR	1
Location of fuel Tankage LCG as % of LBP from FP	65%
Location of fuel Tankage VCG as % of Depth	10%

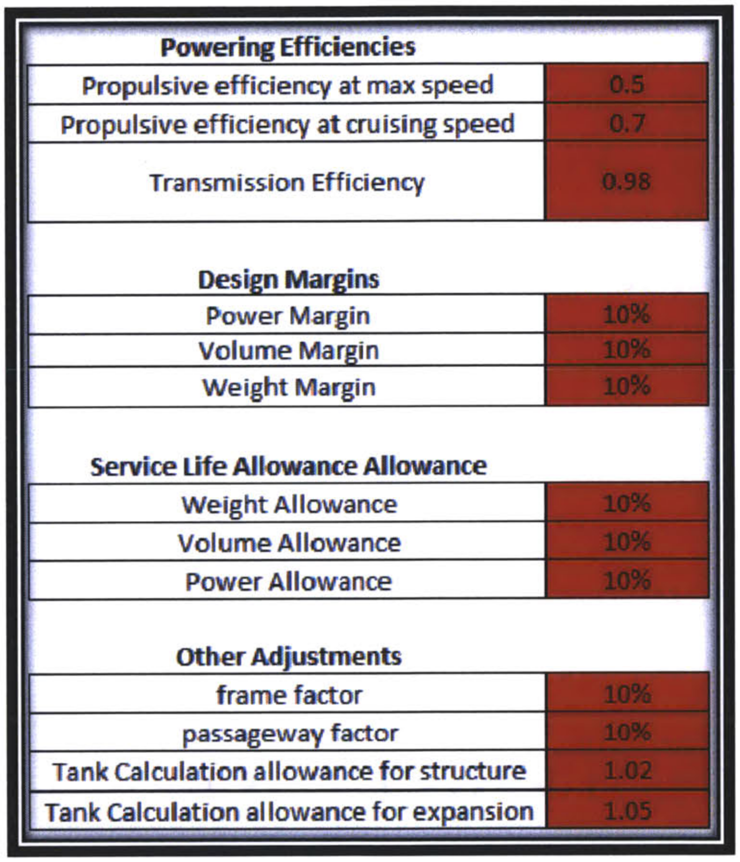
Parametrics

Hotel Load per person (kW)	10
Potable Water per person (LTs)	0.149
Sewage per person (ft^3)	2
fuel per helo per day (LT/helo/day)	1.43

Manning & Stores Defaults

Percentage of officers	10%
CO/XO area req per person (ft^2)	110
officer area req per person (ft^2)	75
enlisted area req per person (ft^2)	50
Minimum Stores (ft^2)	300
stores area density (ft^2/lb)	0.0158
Stores per person (lb/day)	9
Stores period (days)	45
% of payload required for support space	5%
fuel density (lb/gal)	7

Figure 18: Defaults



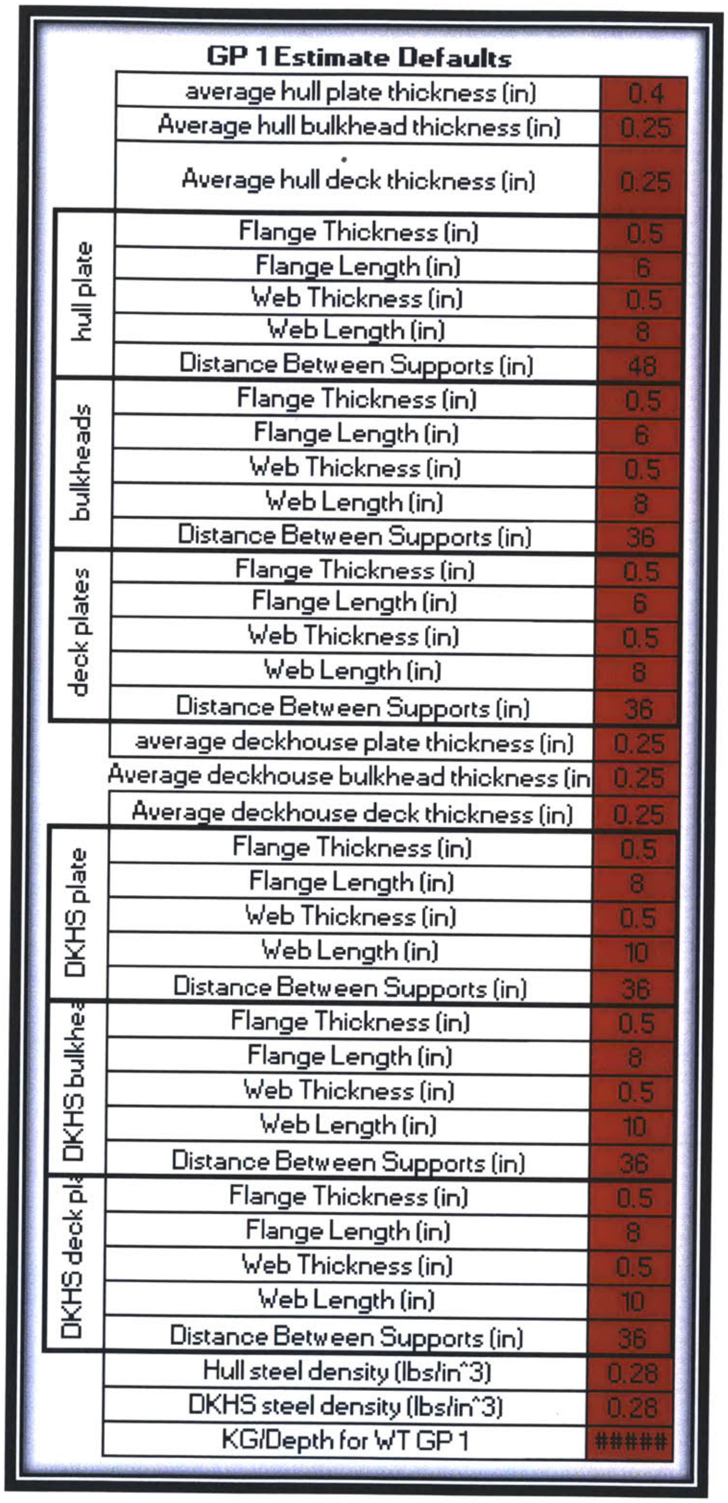
Powering Efficiencies	
Propulsive efficiency at max speed	0.5
Propulsive efficiency at cruising speed	0.7
Transmission Efficiency	0.98

Design Margins	
Power Margin	10%
Volume Margin	10%
Weight Margin	10%

Service Life Allowance Allowance	
Weight Allowance	10%
Volume Allowance	10%
Power Allowance	10%

Other Adjustments	
frame factor	10%
passageway factor	10%
Tank Calculation allowance for structure	1.02
Tank Calculation allowance for expansion	1.05

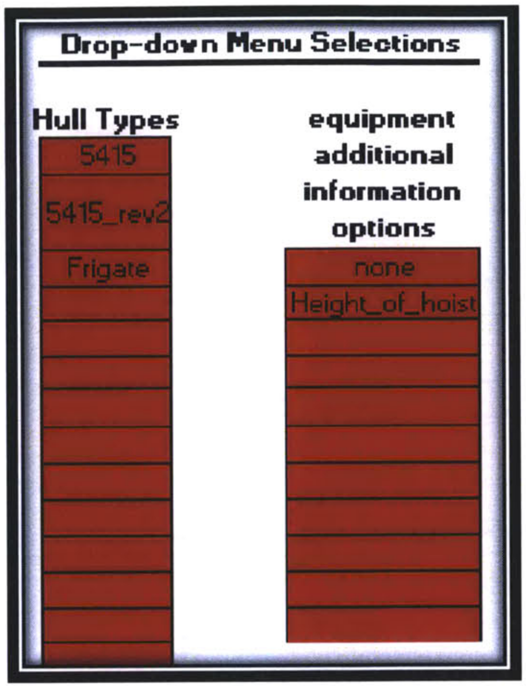
Figure 19: Defaults (cont.)



GP 1 Estimate Defaults

	average hull plate thickness (in)	0.4
	Average hull bulkhead thickness (in)	0.25
	Average hull deck thickness (in)	0.25
hull plate	Flange Thickness (in)	0.5
	Flange Length (in)	6
	Web Thickness (in)	0.5
	Web Length (in)	8
	Distance Between Supports (in)	48
bulkheads	Flange Thickness (in)	0.5
	Flange Length (in)	6
	Web Thickness (in)	0.5
	Web Length (in)	8
	Distance Between Supports (in)	36
deck plates	Flange Thickness (in)	0.5
	Flange Length (in)	6
	Web Thickness (in)	0.5
	Web Length (in)	8
	Distance Between Supports (in)	36
	average deckhouse plate thickness (in)	0.25
	Average deckhouse bulkhead thickness (in	0.25
	Average deckhouse deck thickness (in)	0.25
DKHS plate	Flange Thickness (in)	0.5
	Flange Length (in)	8
	Web Thickness (in)	0.5
	Web Length (in)	10
	Distance Between Supports (in)	36
DKHS bulkhea	Flange Thickness (in)	0.5
	Flange Length (in)	8
	Web Thickness (in)	0.5
	Web Length (in)	10
	Distance Between Supports (in)	36
DKHS deck pl	Flange Thickness (in)	0.5
	Flange Length (in)	8
	Web Thickness (in)	0.5
	Web Length (in)	10
	Distance Between Supports (in)	36
	Hull steel density (lbs/in^3)	0.28
	DKHS steel density (lbs/in^3)	0.28
	KG/Depth for WT GP 1	#####

Figure 20: Defaults (cont.)



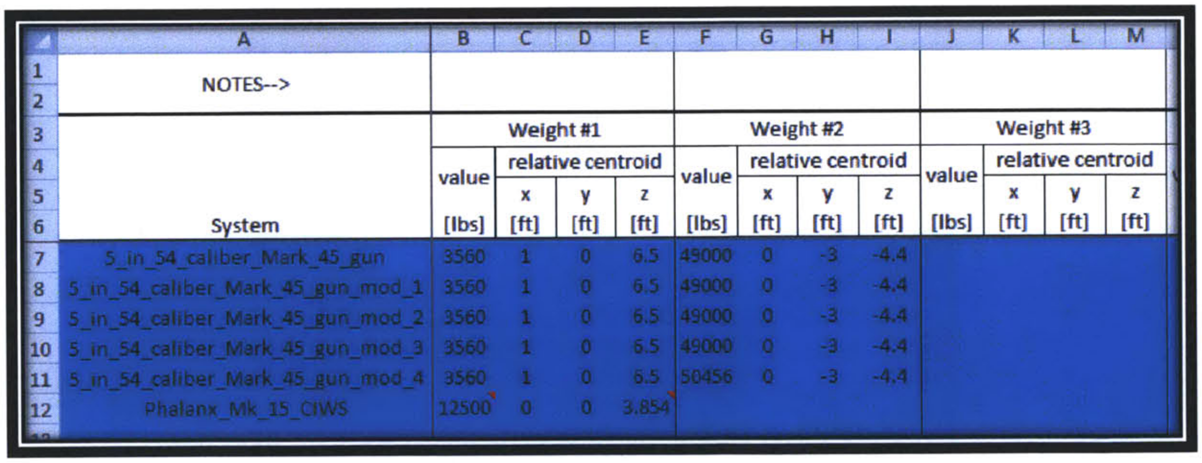
Drop-down Menu Selections

Hull Types	equipment additional information options
5415	none
5415_rev2	Height_of_hoist
Frigate	

Figure 21: Defaults (cont.)

Equipment Libraries:

The equipment libraries are indicated by blue tabs. To add new equipment to the ESSDT, the equipment must be simply added to the respective equipment library. Figure 22 shows the first portion of the "Gun" library.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	NOTES-->												
2													
3		Weight #1				Weight #2				Weight #3			
4		value	relative centroid			value	relative centroid			value	relative centroid		
5			x	y	z		x	y	z		x	y	z
6	System	[lbs]	[ft]	[ft]	[ft]	[lbs]	[ft]	[ft]	[ft]	[lbs]	[ft]	[ft]	[ft]
7	5_in_54_caliber_Mark_45_gun	3560	1	0	6.5	49000	0	-3	-4.4				
8	5_in_54_caliber_Mark_45_gun_mod_1	3560	1	0	6.5	49000	0	-3	-4.4				
9	5_in_54_caliber_Mark_45_gun_mod_2	3560	1	0	6.5	49000	0	-3	-4.4				
10	5_in_54_caliber_Mark_45_gun_mod_3	3560	1	0	6.5	49000	0	-3	-4.4				
11	5_in_54_caliber_Mark_45_gun_mod_4	3560	1	0	6.5	50456	0	-3	-4.4				
12	Phalanx_Mk_15_CIWS	12500	0	0	3.854								

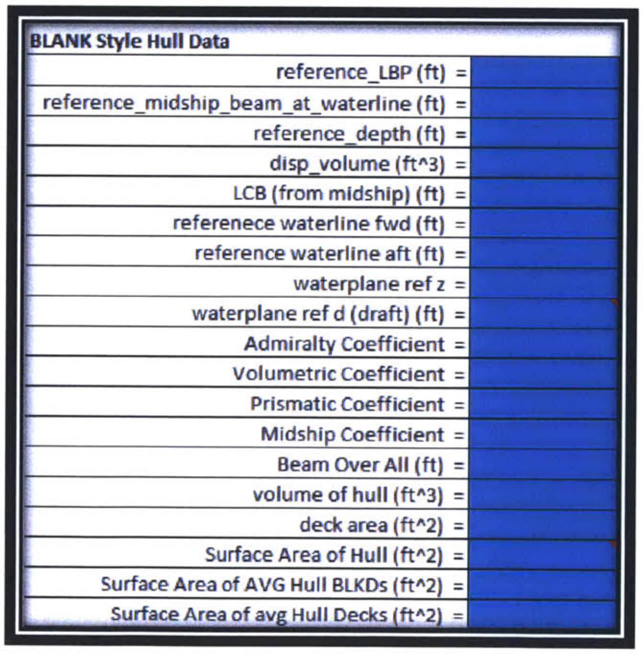
Figure 22: Start of Equipment Library

The unique name for newly added equipment should be added under the column heading "system". This name must contain no spaces and be constructed from only alpha-numerics and underscores. After that, the equipment's specifications may be entered under their respective columns. The library can accommodate up to four different weights, clearance boxes, graphics, and electrical requirements for each piece of equipment. If a graphic is designated in slot number one, then it will take the place of clearance box number one in the Paramarine model. The same is true for graphic number two in relation to clearance box number two and so on.

Hull Libraries:

Adding an entry to the hull library is a little more complex than adding equipment. First, the tab "Blank_Hull_Library" should be copied and renamed to whatever the new hull is to be designated. The new name should contain no spaces or unusual symbols. The new name must also be added to the "Hull Types" list on the "Defined_Defaults" tab as shown in Figure 21.

To add a parent hull to the library, first the parameters shown in Figure 23 should be entered.



BLANK Style Hull Data	
reference_LBP (ft) =	
reference_midship_beam_at_waterline (ft) =	
reference_depth (ft) =	
disp_volume (ft^3) =	
LCB (from midship) (ft) =	
referenece waterline fwd (ft) =	
reference waterline aft (ft) =	
waterplane ref z =	
waterplane ref d (draft) (ft) =	
Admiralty Coefficient =	
Volumetric Coefficient =	
Prismatic Coefficient =	
Midship Coefficient =	
Beam Over All (ft) =	
volume of hull (ft^3) =	
deck area (ft^2) =	
Surface Area of Hull (ft^2) =	
Surface Area of AVG Hull BLKDs (ft^2) =	
Surface Area of avg Hull Decks (ft^2) =	

Figure 23: Initial Inputs for an Addition to the Hull Library

The rest of the hull is comprised of nine Key Points and seven Guide Curves. The key points represent the top and bottom of the transom, aft portion of the Parallel Mid-Body (PMB), forward portion of the PMB, and the bow as well as the point of the aft cut-up as illustrated in Figure 24. Since most warships do not have a PMB, the points for the aft and forward portions of the PMB are typically concurrent. The key points are entered in the table illustrated in Figure 25.

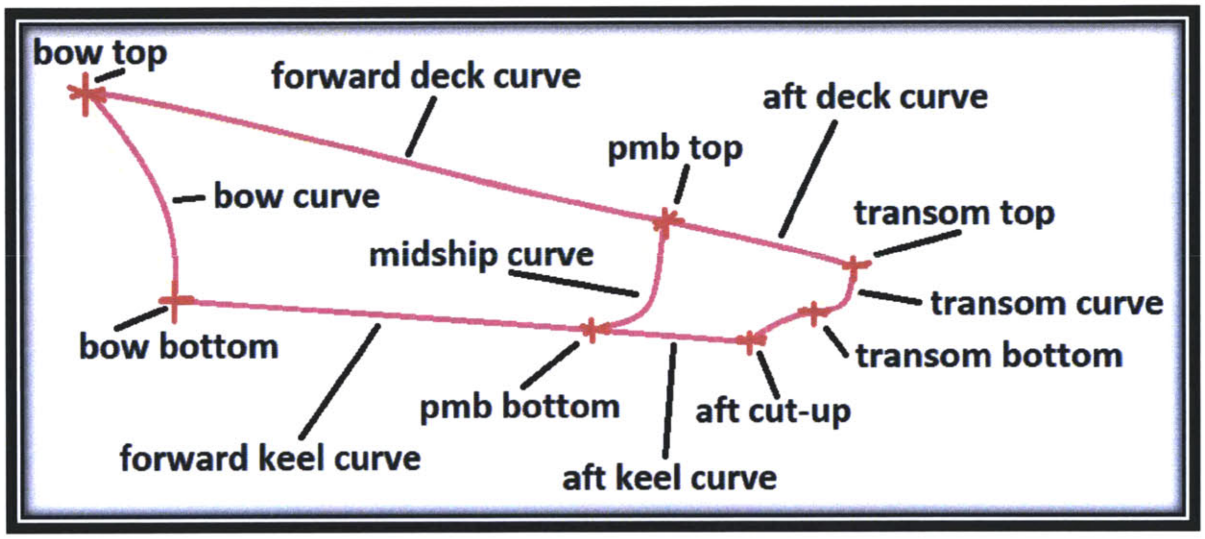
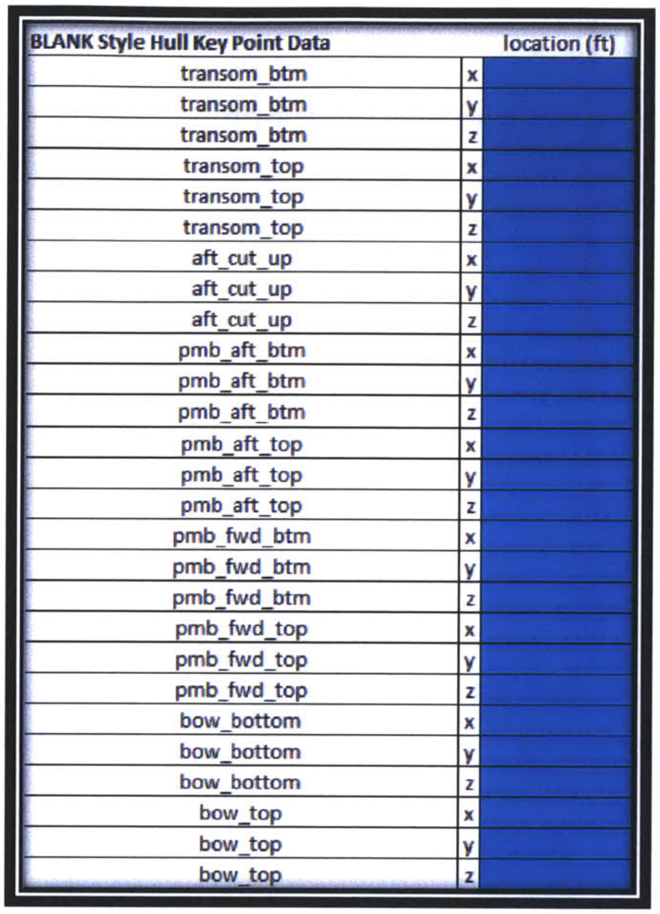


Figure 24: Illustration of Guidance Geometry for the Hull Library



BLANK Style Hull Key Point Data		location (ft)
transom_btm	x	
transom_btm	y	
transom_btm	z	
transom_top	x	
transom_top	y	
transom_top	z	
aft_cut_up	x	
aft_cut_up	y	
aft_cut_up	z	
pmb_aft_btm	x	
pmb_aft_btm	y	
pmb_aft_btm	z	
pmb_aft_top	x	
pmb_aft_top	y	
pmb_aft_top	z	
pmb_fwd_btm	x	
pmb_fwd_btm	y	
pmb_fwd_btm	z	
pmb_fwd_top	x	
pmb_fwd_top	y	
pmb_fwd_top	z	
bow_bottom	x	
bow_bottom	y	
bow_bottom	z	
bow_top	x	
bow_top	y	
bow_top	z	

Figure 25: Table for Entry of Key Points for the Hull Library

Figure 24 also illustrates the seven guide curves. These guide curves are comprised of eight points. Points one and eight are the ends of the curve. The other six points should be selected such that they define the curve. Through relatively straight sections of the curve, the points can be selected fairly far apart. They should be grouped closer through sections of greater curvature. So, for example, in Figure 24, the midship curve should have the majority of its points for definition where it transitions from its fairly flat bottom to fairly flat side, as typified in Figure 26. These points are entered into the table shown in Figure 27.

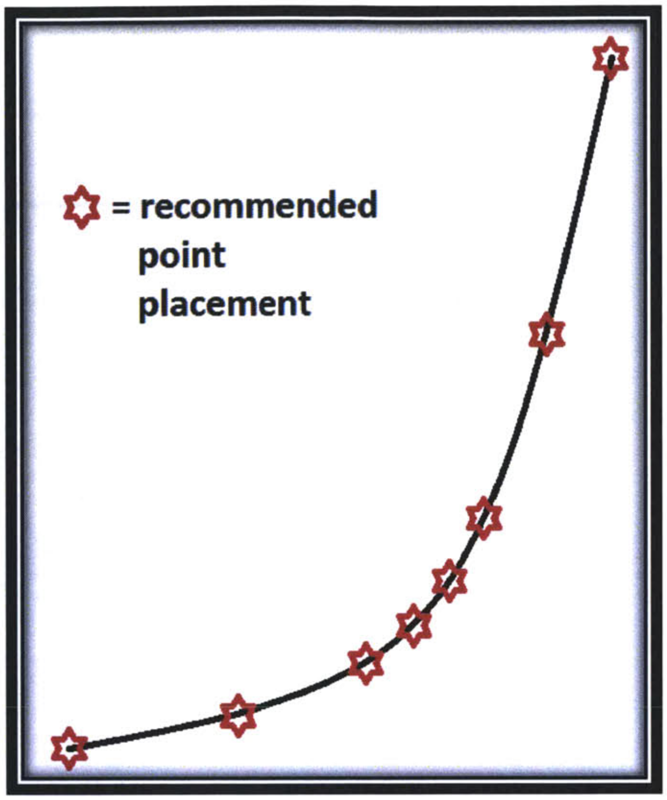
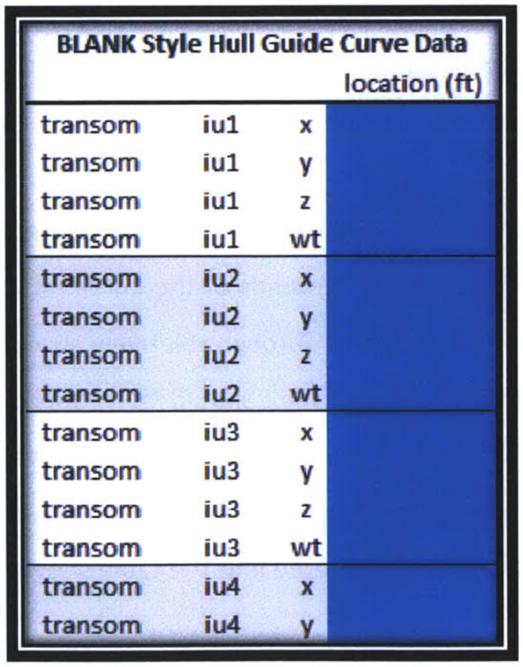


Figure 26: Example Point Placement for a Typical Midship Curve

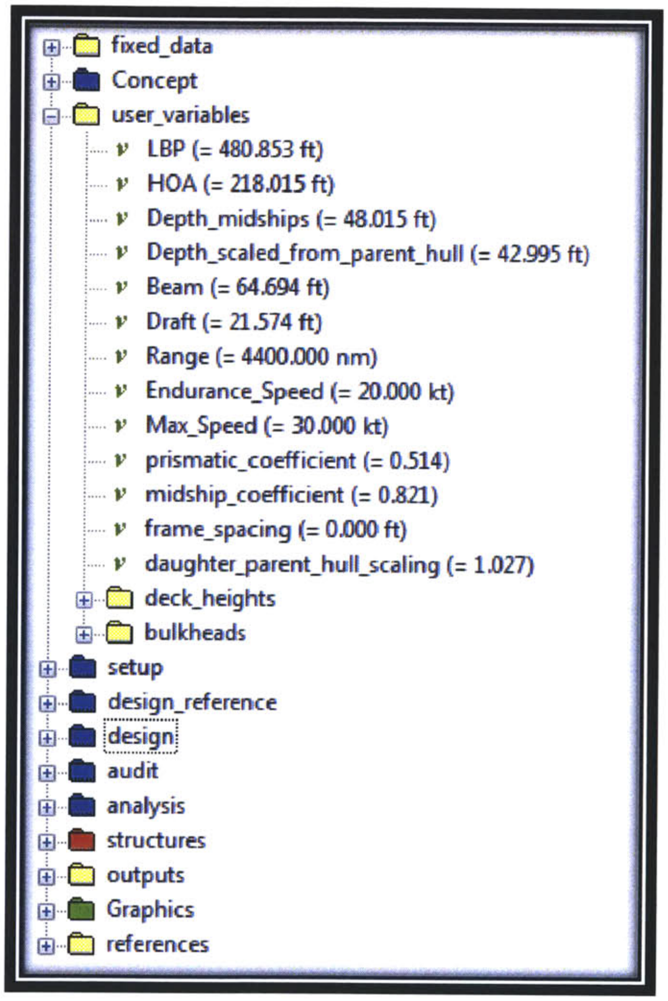


BLANK Style Hull Guide Curve Data			location (ft)
transom	iu1	x	
transom	iu1	y	
transom	iu1	z	
transom	iu1	wt	
transom	iu2	x	
transom	iu2	y	
transom	iu2	z	
transom	iu2	wt	
transom	iu3	x	
transom	iu3	y	
transom	iu3	z	
transom	iu3	wt	
transom	iu4	x	
transom	iu4	y	

Figure 27: First Few Rows of Table for Guide Curve Control Point Entry

Paramarine:

Once imported into Paramarine, the model is ready for further analysis and development. Figure 28 displays where different key dimensions and parameters of the ship may be altered in the model.



```
fixed_data
Concept
user_variables
    LBP (= 480.853 ft)
    HOA (= 218.015 ft)
    Depth_midships (= 48.015 ft)
    Depth_scaled_from_parent_hull (= 42.995 ft)
    Beam (= 64.694 ft)
    Draft (= 21.574 ft)
    Range (= 4400.000 nm)
    Endurance_Speed (= 20.000 kt)
    Max_Speed (= 30.000 kt)
    prismatic_coefficient (= 0.514)
    midship_coefficient (= 0.821)
    frame_spacing (= 0.000 ft)
    daughter_parent_hull_scaling (= 1.027)
    deck_heights
    bulkheads
setup
design_reference
design
audit
analysis
structures
outputs
Graphics
references
```

Figure 28: User Variables

The physical model is located in the design folder. The hull and bulkheads are placed as shown in Figure 29. The rest of the equipment and deckhouse is located in the "design_building_block_model" within the "design" folder.

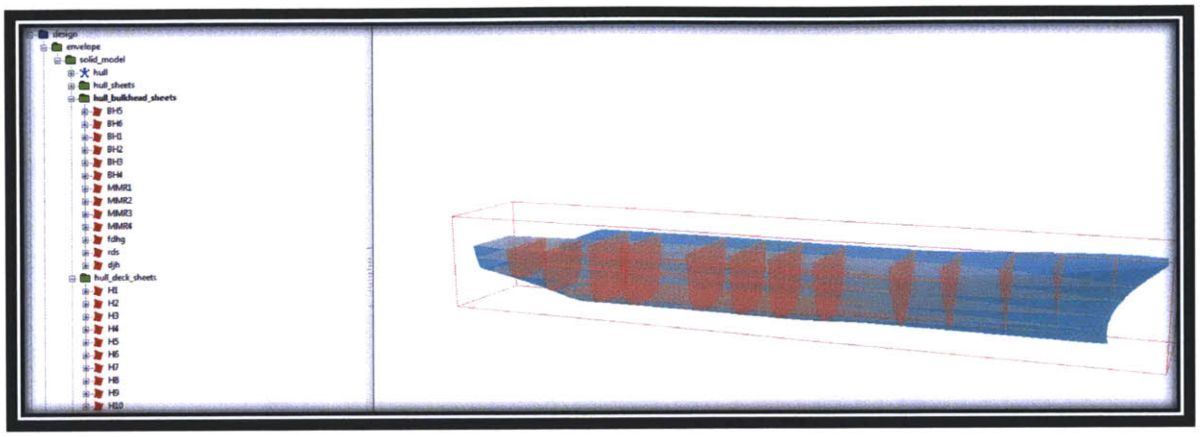


Figure 29: Hull, Decks, and Bulkheads

Powering Analysis can be found in the "analysis" folder under "powering", "2_analysis_deep", "visualiser".

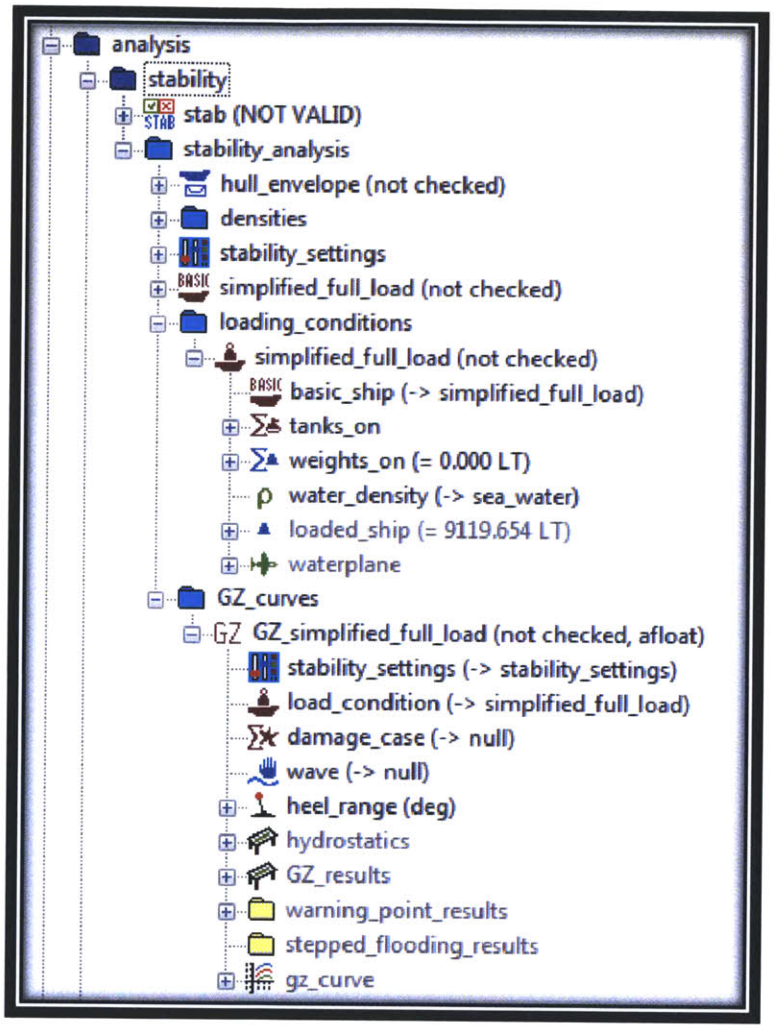
For further analysis, the following links must be made manually:

Set design/design_building_block_model/Design/requirements/fuel/characteristics/propulsion_power_at_endurance_speed equal to the power under analysis/powering/visualiser/tabular_results/shaft_power which corresponds to the endurance speed for the modeled vessel.

Set audit/weight/margins/lightship equal to audit/weight/lightship/design_audit/results/Total/weight.

Set the weight and centroid of analysis/stability/stability_analysis/simplified_full_load/datum equal to the weight and x, y, and z centroids of audit/weight/simplified_full_load/design_audit/results/Total.

Once these links are accomplished the Paramarine model is also able to perform a preliminary trim and stability analysis which is located in analysis/stability/stability_analysis as shown in Figure 30.



```
analysis
  stability
    stab (NOT VALID)
    stability_analysis
      hull_envelope (not checked)
      densities
      stability_settings
      simplified_full_load (not checked)
      loading_conditions
        simplified_full_load (not checked)
          basic_ship (-> simplified_full_load)
          tanks_on
          weights_on (= 0.000 LT)
          water_density (-> sea_water)
          loaded_ship (= 9119.654 LT)
          waterplane
      GZ_curves
        GZ_simplified_full_load (not checked, afloat)
          stability_settings (-> stability_settings)
          load_condition (-> simplified_full_load)
          damage_case (-> null)
          wave (-> null)
          heel_range (deg)
          hydrostatics
          GZ_results
          warning_point_results
          stepped_flooding_results
          gz_curve
```

Figure 30: Stability Analysis

This process also inserts the model into Paramarine in a manner similar to the file structure outlined in *Surface Ship Early Stage Design Training Course for Warship Design* (Pawling, 2009) such that is is ready for further analysis andd development within the Paramarine program.

(THIS PAGE INTENTIONALLY LEFT BLANK)

Appendix B: Examples of ESSDT Entry, Analysis, and Output Worksheets

Defined Defaults Tab:

Geometric Defaults	
min L/B	6.5
max L/B	10
fore/aft split point for graphic orientation det. (% aft from FP)	50%
Inner Bottom Height Above Baseline	59%
Hull Deck Height for deck above MMR(ft)	8.5
Hull Deck Height for 1 deck above MMR(ft)	8.5
Hull Deck Height for 2 decks above MMR(ft)	8.5
Hull Deck Height for 3 decks above MMR(ft)	8.5
Deckhouse Level Height (ft)	9
# of Hull Decks above MMR	1
Location of fuel Tankage LCG as % of LBP from FP	65%
Location of fuel Tankage VCG as % of Depth	10%

Parametrics	
Hotel Load per person (kW)	10
Potable Water per person (LTs)	0.149
Sewage per person (ft^3)	2
fuel per helo per day (LT/helo/day)	1.43

Manning & Stores Defaults	
Percentage of officers	10%
CO/XO area req per person (ft^2)	110
officer area req per person (ft^2)	75
enlisted area req per person (ft^2)	50
Minimum Stores (ft^2)	300
stores area density (ft^2/lb)	0.0158
Stores per person (lb/day)	9
Stores period (days)	45
% of payload required for support space	5%
fuel density (lb/gal)	7

Powering Efficiencies	
Propulsive efficiency at max speed	0.5
Propulsive efficiency at cruising speed	0.7
Transmission Efficiency	0.98

Design Margins	
Power Margin	10%
Volume Margin	10%
Weight Margin	10%

Service Life Allowance Allowance	
Weight Allowance	10%
Volume Allowance	10%
Power Allowance	10%

Other Adjustments	
frame factor	10%
passageway factor	10%
Tank Calculation allowance for structure	1.02
Tank Calculation allowance for expansion	1.05

GP 1 Estimate Defaults

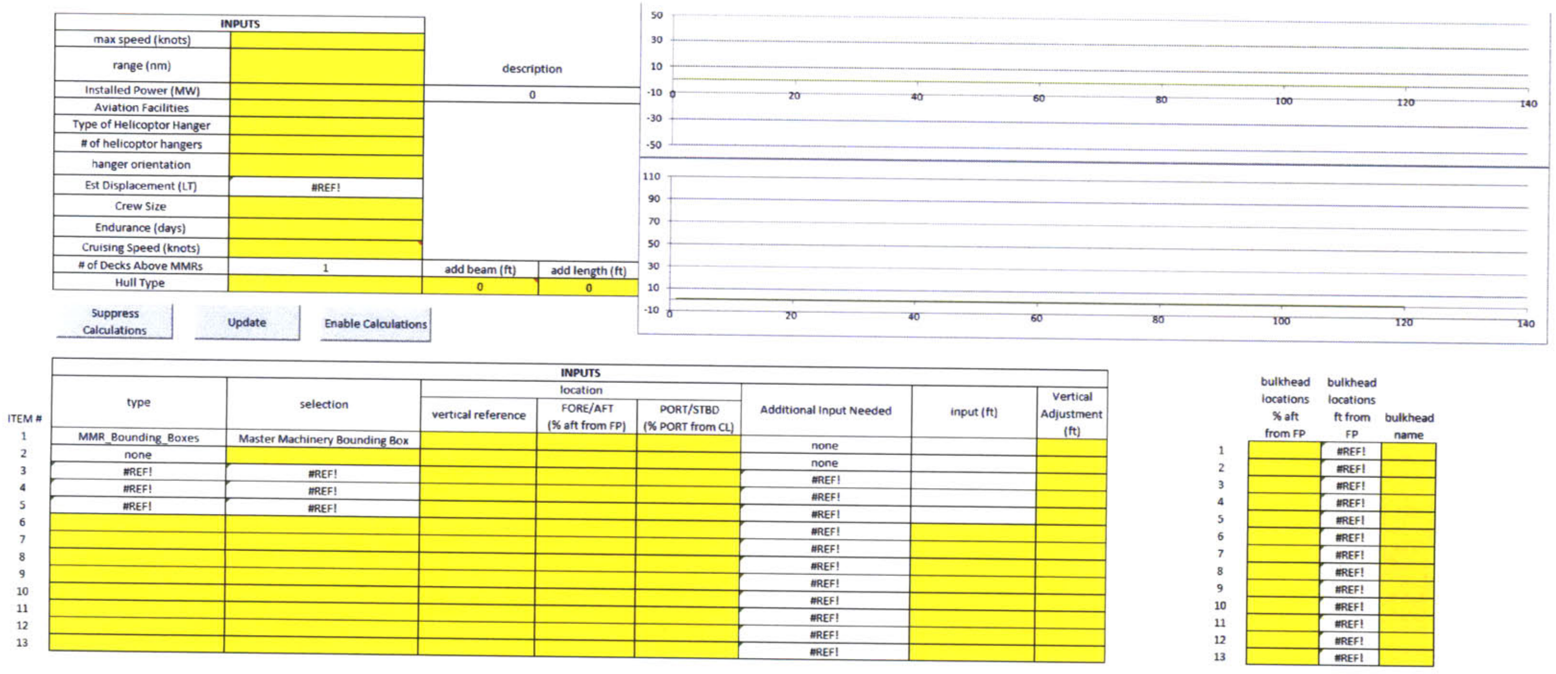
	average hull plate thickness (in)	0.4
	Average hull bulkhead thickness (in)	0.25
	Average hull deck thickness (in)	0.25
hull plate	Flange Thickness (in)	0.5
	Flange Length (in)	6
	Web Thickness (in)	0.5
	Web Length (in)	8
	Distance Between Supports (in)	48
bulkheads	Flange Thickness (in)	0.5
	Flange Length (in)	6
	Web Thickness (in)	0.5
	Web Length (in)	8
	Distance Between Supports (in)	36
deck plates	Flange Thickness (in)	0.5
	Flange Length (in)	6
	Web Thickness (in)	0.5
	Web Length (in)	8
	Distance Between Supports (in)	36
	average deckhouse plate thickness (in)	0.25
	Average deckhouse bulkhead thickness (in)	0.25
	Average deckhouse deck thickness (in)	0.25
DKHS plate	Flange Thickness (in)	0.5
	Flange Length (in)	8
	Web Thickness (in)	0.5
	Web Length (in)	10
	Distance Between Supports (in)	36
DKHS bulkheads	Flange Thickness (in)	0.5
	Flange Length (in)	8
	Web Thickness (in)	0.5
	Web Length (in)	10
	Distance Between Supports (in)	36
DKHS deck plate	Flange Thickness (in)	0.5
	Flange Length (in)	8
	Web Thickness (in)	0.5
	Web Length (in)	10
	Distance Between Supports (in)	36
	Hull steel density (lbs/in^3)	0.28
	DKHS steel density (lbs/in^3)	0.28
	KG/Depth for WT GP 1	61.71%

Drop-down Menu Selections

Hull Types
5415
5415_rev2
Frigate

equipment additional information options
none
Height_of_hoist

Initial Input Tab:



INPUTS			
max speed (knots)			
range (nm)		description	
Installed Power (MW)		0	
Aviation Facilities			
Type of Helicoptor Hanger			
# of helicoptor hangers			
hanger orientation			
Est Displacement (LT)	#REF!		
Crew Size			
Endurance (days)			
Cruising Speed (knots)			
# of Decks Above MMRs	1	add beam (ft)	add length (ft)
Hull Type		0	0

Suppress Calculations

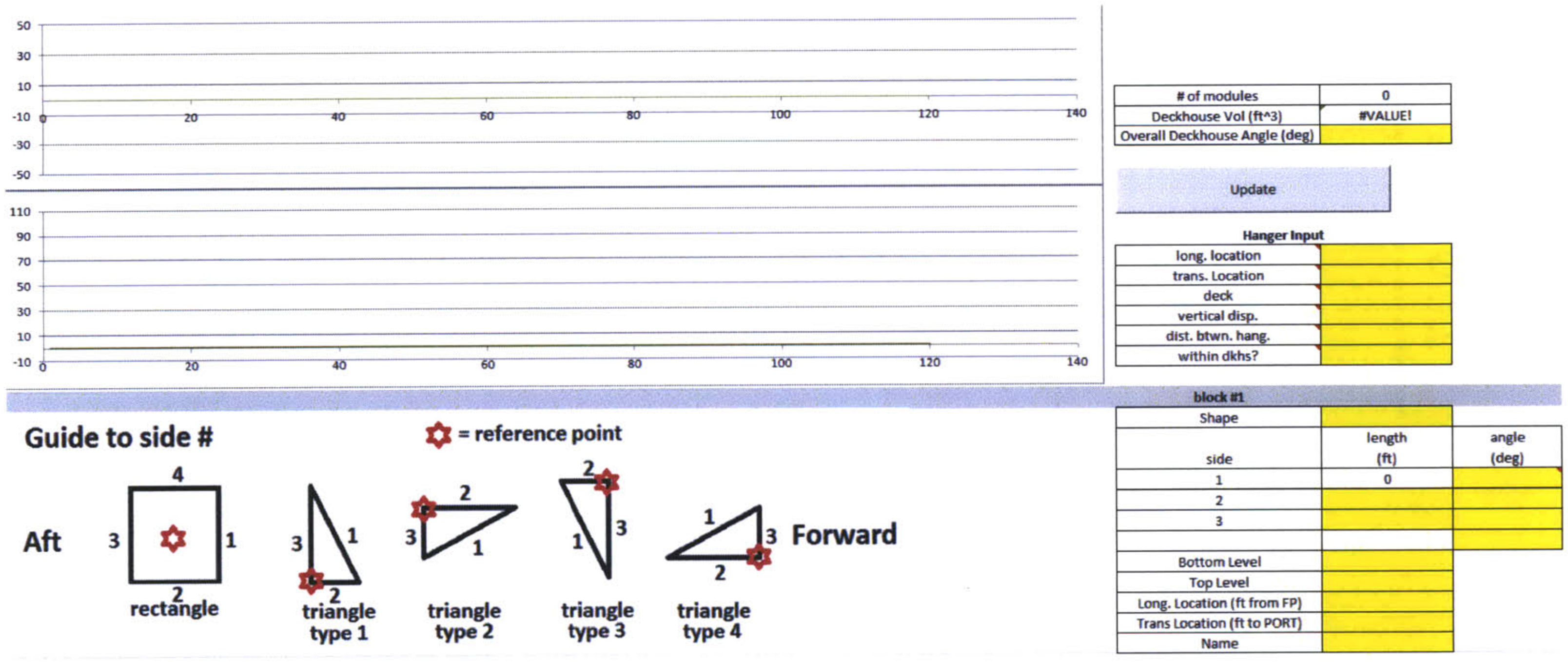
Update

Enable Calculations

	INPUTS							
			location					
ITEM #	type	selection	vertical reference	FORE/AFT (% aft from FP)	PORT/STBD (% PORT from CL)	Additional Input Needed	input (ft)	Vertical Adjustment (ft)
1	MMR_Bounding_Boxes	Master Machinery Bounding Box				none		
2	none					none		
3	#REF!	#REF!				#REF!		
4	#REF!	#REF!				#REF!		
5	#REF!	#REF!				#REF!		
6						#REF!		
7						#REF!		
8						#REF!		
9						#REF!		
10						#REF!		
11						#REF!		
12						#REF!		
13						#REF!		

	bulkhead locations % aft from FP	bulkhead locations ft from FP	bulkhead name
1		#REF!	
2		#REF!	
3		#REF!	
4		#REF!	
5		#REF!	
6		#REF!	
7		#REF!	
8		#REF!	
9		#REF!	
10		#REF!	
11		#REF!	
12		#REF!	
13		#REF!	

Create Deckhouse Tab:



# of modules	0
Deckhouse Vol (ft^3)	#VALUE!
Overall Deckhouse Angle (deg)	

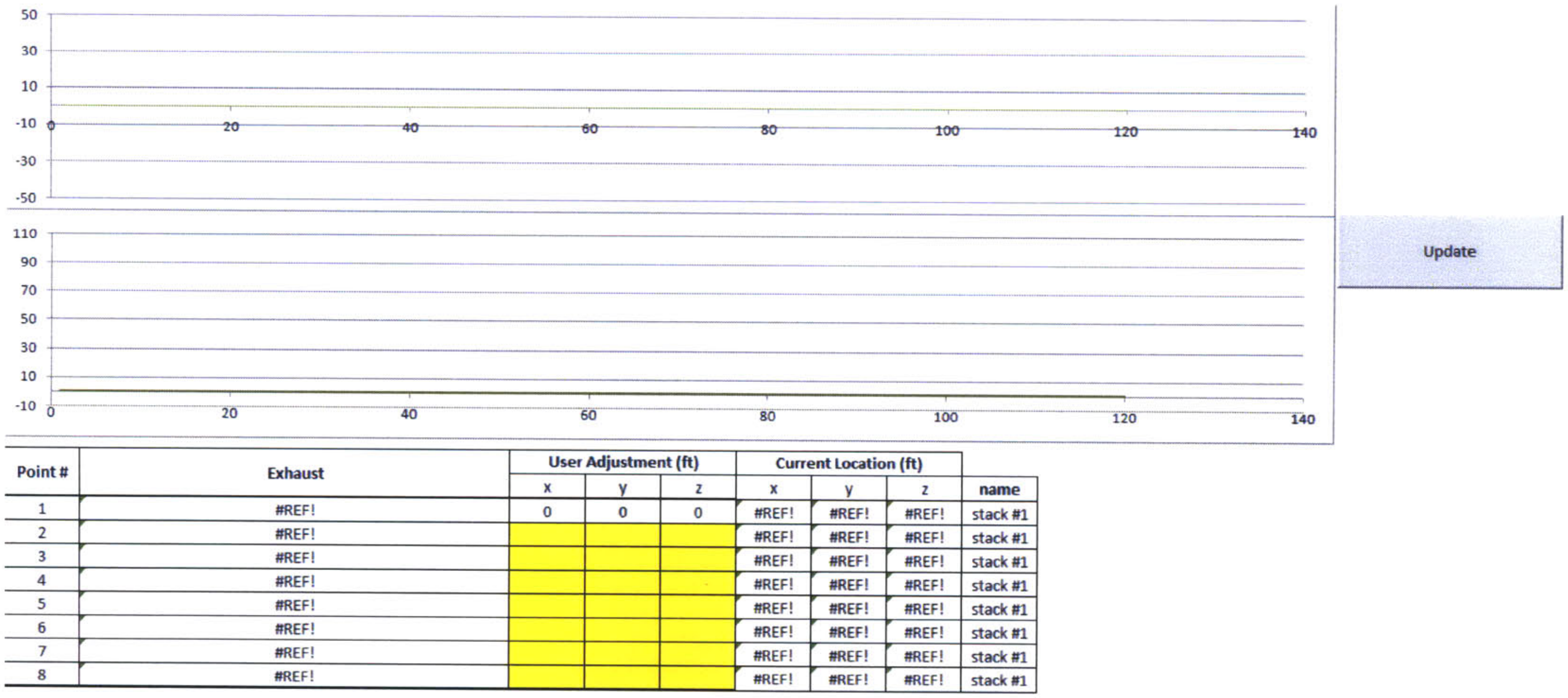
Hanger Input

long. location	
trans. Location	
deck	
vertical disp.	
dist. btwn. hang.	
within dkhs?	

block #1

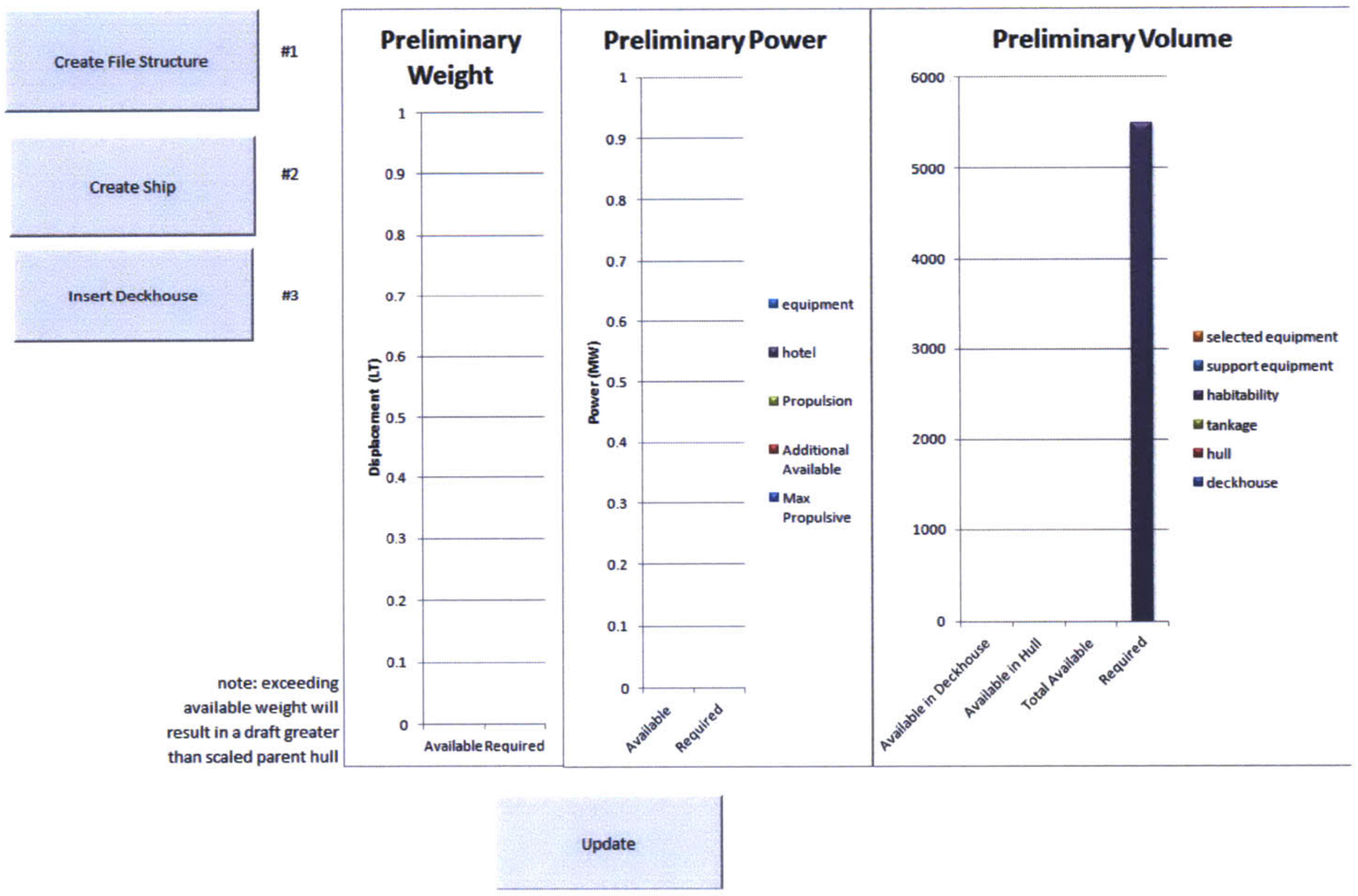
Shape		
side	length (ft)	angle (deg)
1	0	
2		
3		
Bottom Level		
Top Level		
Long. Location (ft from FP)		
Trans Location (ft to PORT)		
Name		

Adjust Stacks Tab:



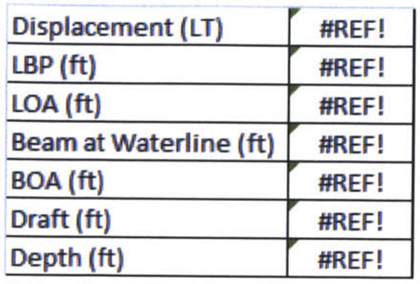
Point #	Exhaust	User Adjustment (ft)			Current Location (ft)			
		x	y	z	x	y	z	name
1	#REF!	0	0	0	#REF!	#REF!	#REF!	stack #1
2	#REF!				#REF!	#REF!	#REF!	stack #1
3	#REF!				#REF!	#REF!	#REF!	stack #1
4	#REF!				#REF!	#REF!	#REF!	stack #1
5	#REF!				#REF!	#REF!	#REF!	stack #1
6	#REF!				#REF!	#REF!	#REF!	stack #1
7	#REF!				#REF!	#REF!	#REF!	stack #1
8	#REF!				#REF!	#REF!	#REF!	stack #1

Import to Paramarine Tab:



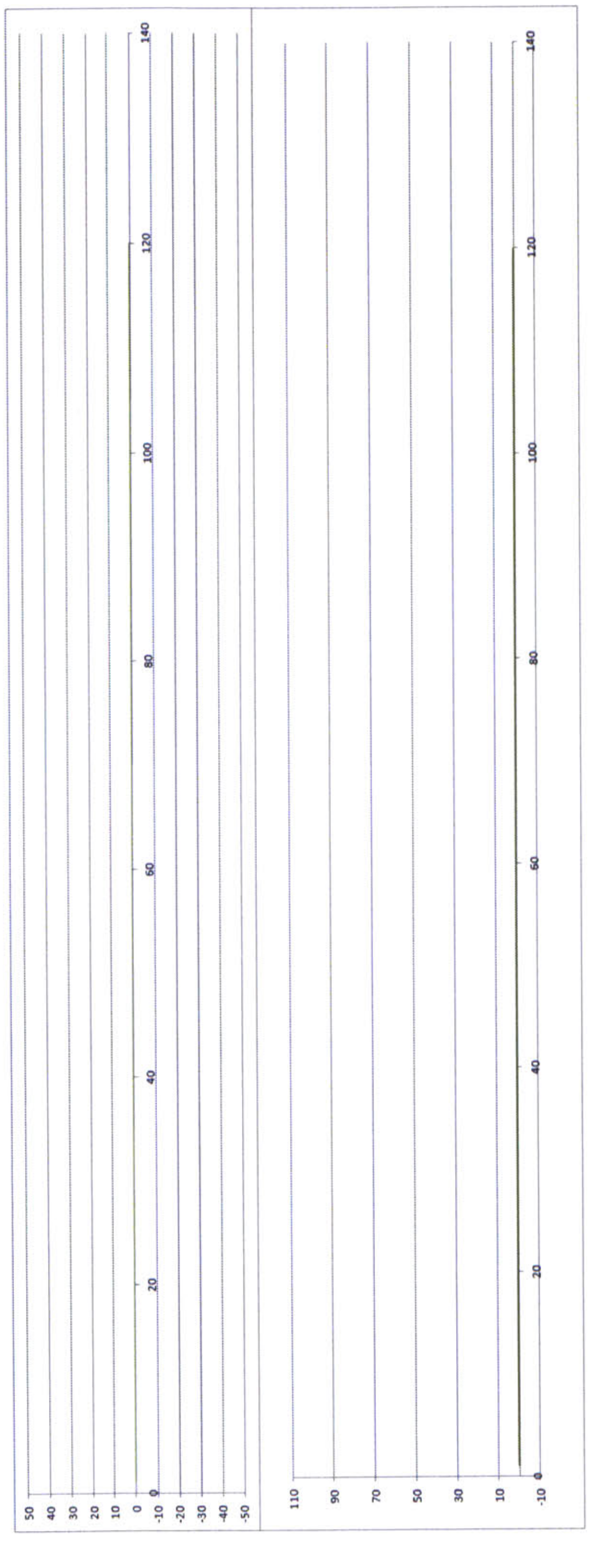
note: exceeding available weight will result in a draft greater than scaled parent hull

Final Output Tab:



Displacement (LT)	#REF!
LBP (ft)	#REF!
LOA (ft)	#REF!
Beam at Waterline (ft)	#REF!
BOA (ft)	#REF!
Draft (ft)	#REF!
Depth (ft)	#REF!

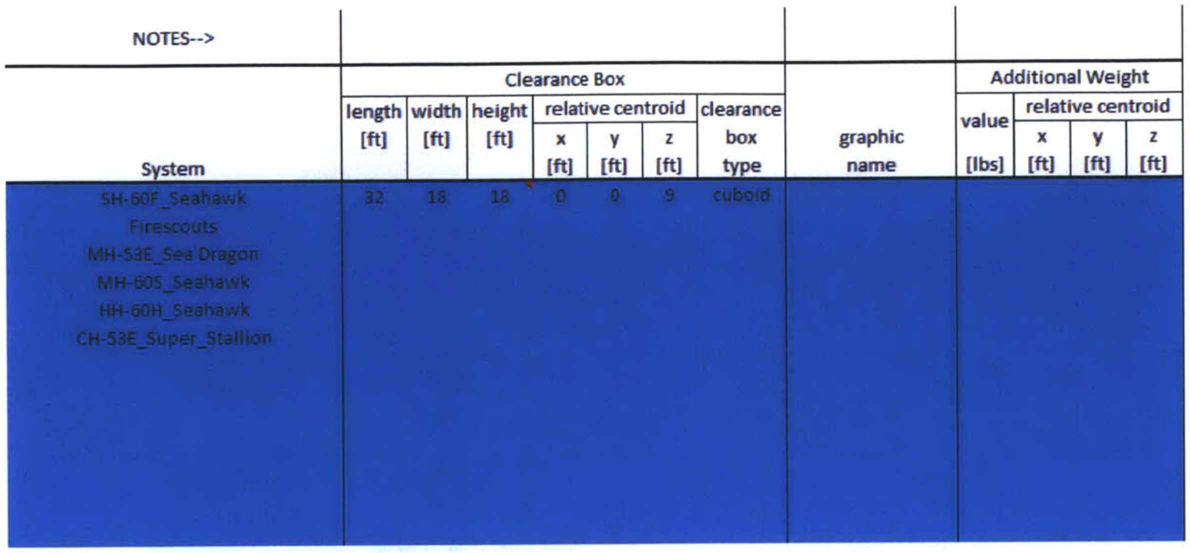
Final Graphic Tab:



Gun Library Tab (first 24 columns):

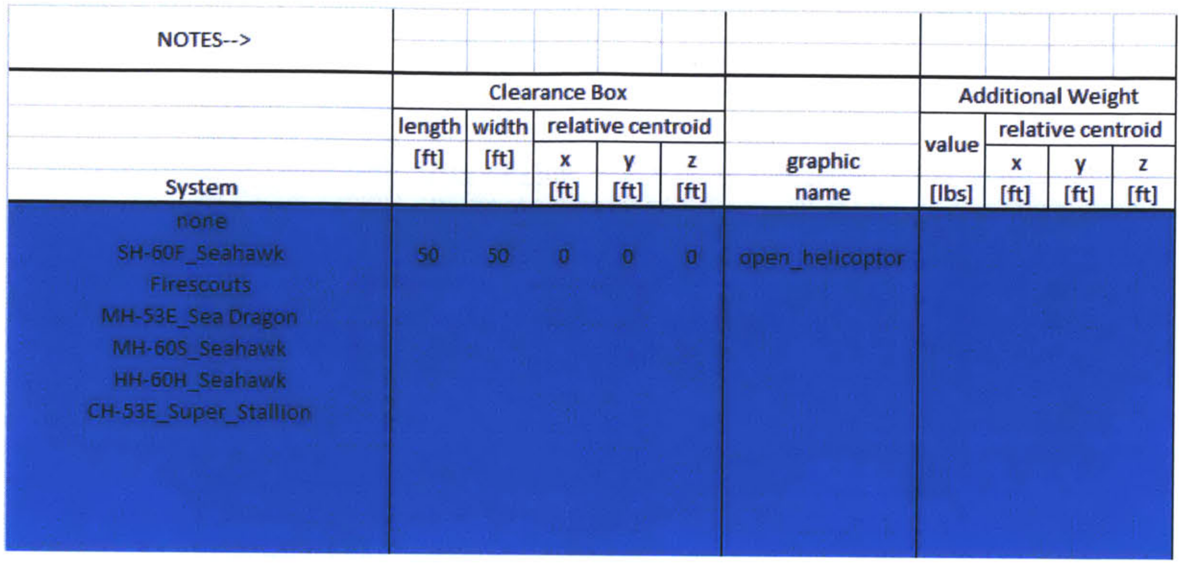
NOTES-->																							
	Weight #1				Weight #2				Weight #3				Weight #4				Clearance Box #1						
	value	relative centroid			value	relative centroid			value	relative centroid			value	relative centroid			length	width	height	relative centroid			clearance box type
		x	y	z		x	y	z		x	y	z		x	y	z				x	y	z	
System	[lbs]	[ft]	[ft]	[ft]	[lbs]	[ft]	[ft]	[ft]	[lbs]	[ft]	[ft]	[ft]	[lbs]	[ft]	[ft]	[ft]	[ft]	[ft]	[ft]	[ft]	[ft]	[ft]	
5_in_54_caliber_Mark_45_gun	3560	1	0	6.5	49000	0	-3	-4.4									44	15.64	10.372	8	0	5.186	cylinder_vert
5_in_54_caliber_Mark_45_gun_mod_1	3560	1	0	6.5	49000	0	-3	-4.4															
5_in_54_caliber_Mark_45_gun_mod_2	3560	1	0	6.5	49000	0	-3	-4.4															
5_in_54_caliber_Mark_45_gun_mod_3	3560	1	0	6.5	49000	0	-3	-4.4															
5_in_54_caliber_Mark_45_gun_mod_4	3560	1	0	6.5	50456	0	-3	-4.4															
Phalanx_Mk_15_CIWS	12500	0	0	3.854													6.1667	6.167	15.417	0	0	7.708	cylinder_vert

Hanger Library Tab:



NOTES-->												
	Clearance Box								Additional Weight			
	length	width	height	relative centroid			clearance		value	relative centroid		
	[ft]	[ft]	[ft]	x	y	z	box	graphic		x	y	z
System				[ft]	[ft]	[ft]	type	name	[lbs]	[ft]	[ft]	[ft]
SH-60F_Seahawk	32	18	18	0	0	9	cuboid					
Firescouts												
MH-53E_Sea Dragon												
MH-60S_Seahawk												
HH-60H_Seahawk												
CH-53E_Super_Stallion												

Flight Deck Library Tab:



NOTES-->										
	Clearance Box						Additional Weight			
	length	width	relative centroid				value	relative centroid		
	[ft]	[ft]	x	y	z	graphic		x	y	z
System			[ft]	[ft]	[ft]	name	[lbs]	[ft]	[ft]	[ft]
none										
SH-60F_Seahawk	50	50	0	0	0	open_helicoptor				
Firescouts										
MH-53E_Sea Dragon										
MH-60S_Seahawk										
HH-60H_Seahawk										
CH-53E_Super_Stallion										

Blank Hull Library Tab:

BLANK Style Hull Data

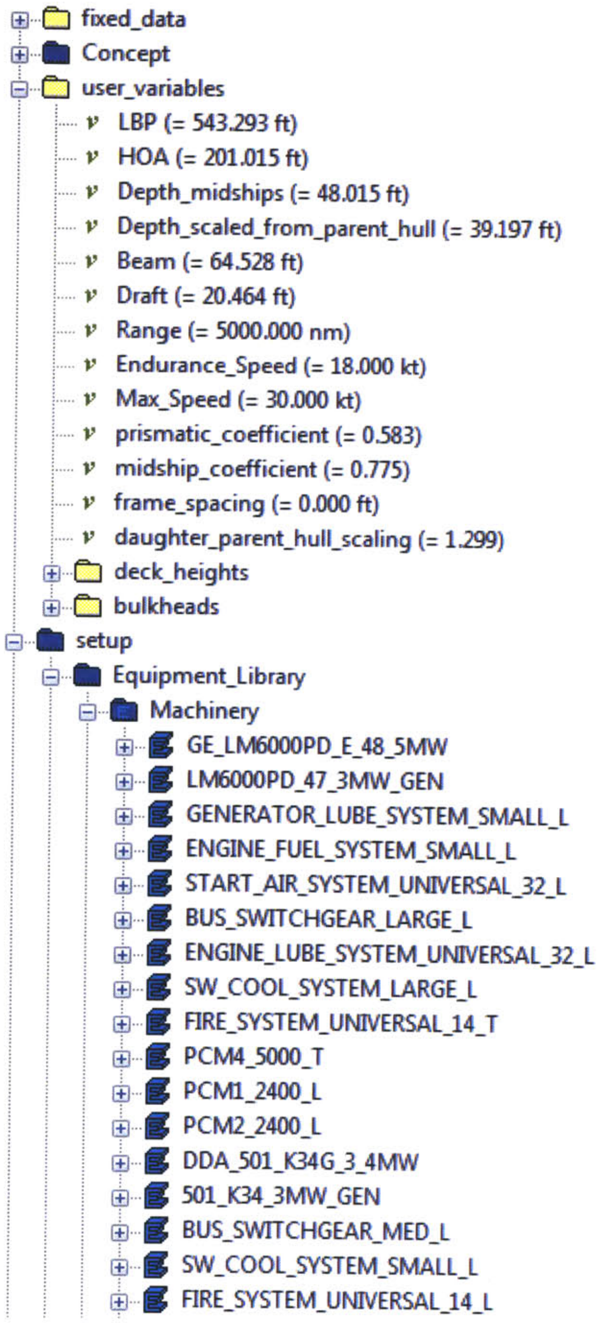
reference_LBP (ft) =	
reference_midship_beam_at_waterline (ft) =	
reference_depth (ft) =	
disp_volume (ft^3) =	
LCB (from midship) (ft) =	
referenece waterline fwd (ft) =	
reference waterline aft (ft) =	
waterplane ref z =	
waterplane ref d (draft) (ft) =	
Admiralty Coefficient =	
Volumetric Coefficient =	
Prismatic Coefficient =	
Midship Coefficient =	
Beam Over All (ft) =	
volume of hull (ft^3) =	
deck area (ft^2) =	
Surface Area of Hull (ft^2) =	
Surface Area of AVG Hull BLKDs (ft^2) =	
Surface Area of avg Hull Decks (ft^2) =	

BLANK Style Hull Key Point Data		location (ft)
transom_btm	x	
transom_btm	y	
transom_btm	z	
transom_top	x	
transom_top	y	
transom_top	z	
aft_cut_up	x	
aft_cut_up	y	
aft_cut_up	z	
pmb_aft_btm	x	
pmb_aft_btm	y	
pmb_aft_btm	z	
pmb_aft_top	x	
pmb_aft_top	y	
pmb_aft_top	z	
pmb_fwd_btm	x	
pmb_fwd_btm	y	
pmb_fwd_btm	z	
pmb_fwd_top	x	
pmb_fwd_top	y	
pmb_fwd_top	z	
bow_bottom	x	
bow_bottom	y	
bow_bottom	z	
bow_top	x	
bow_top	y	
bow_top	z	

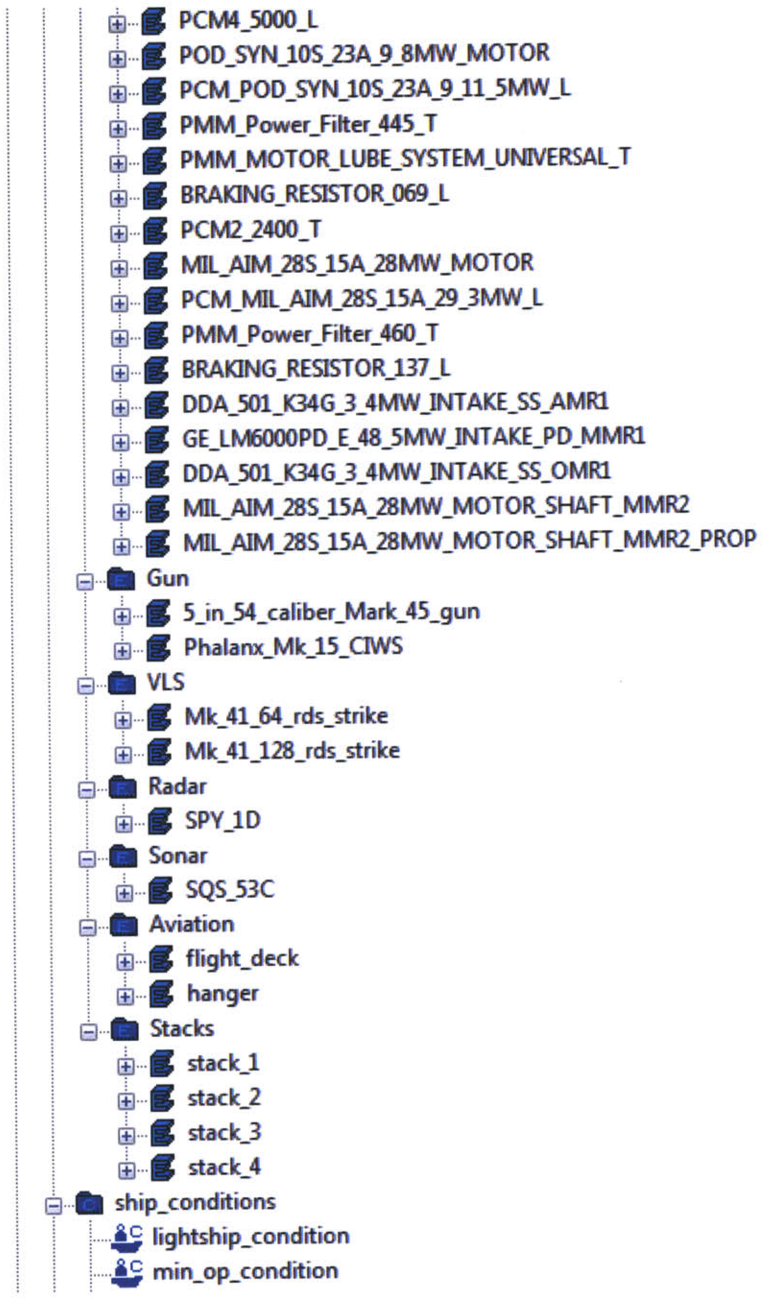
BLANK Style Hull Guide Curve Data

			location (ft)
transom	iu1	x	
transom	iu1	y	
transom	iu1	z	
transom	iu1	wt	
transom	iu2	x	
transom	iu2	y	
transom	iu2	z	
transom	iu2	wt	
transom	iu3	x	
transom	iu3	y	
transom	iu3	z	
transom	iu3	wt	
transom	iu4	x	
transom	iu4	y	
transom	iu4	z	
transom	iu4	wt	
transom	iu5	x	
transom	iu5	y	
transom	iu5	z	
transom	iu5	wt	
transom	iu6	x	
transom	iu6	y	
transom	iu6	z	
transom	iu6	wt	
transom	iu7	x	
transom	iu7	y	
transom	iu7	z	
transom	iu7	wt	
transom	iu8	x	
transom	iu8	y	
transom	iu8	z	
transom	iu8	wt	
midships	iu1	x	
midships	iu1	y	
midships	iu1	z	
midships	iu1	wt	

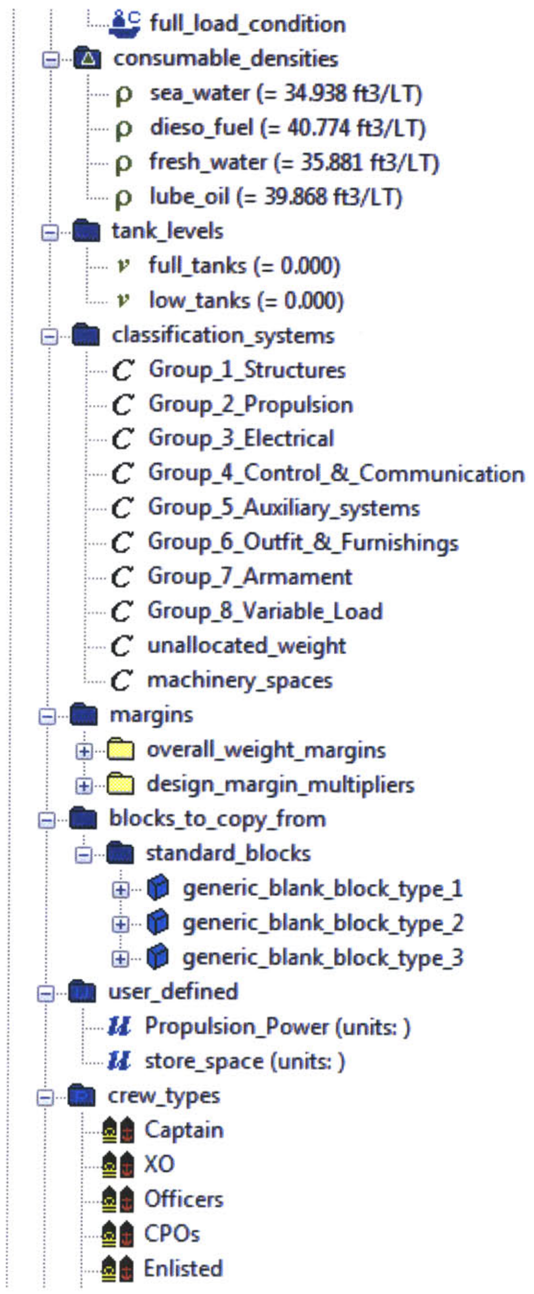
Appendix C: File Structure Created Within Paramarine by the ESSDT



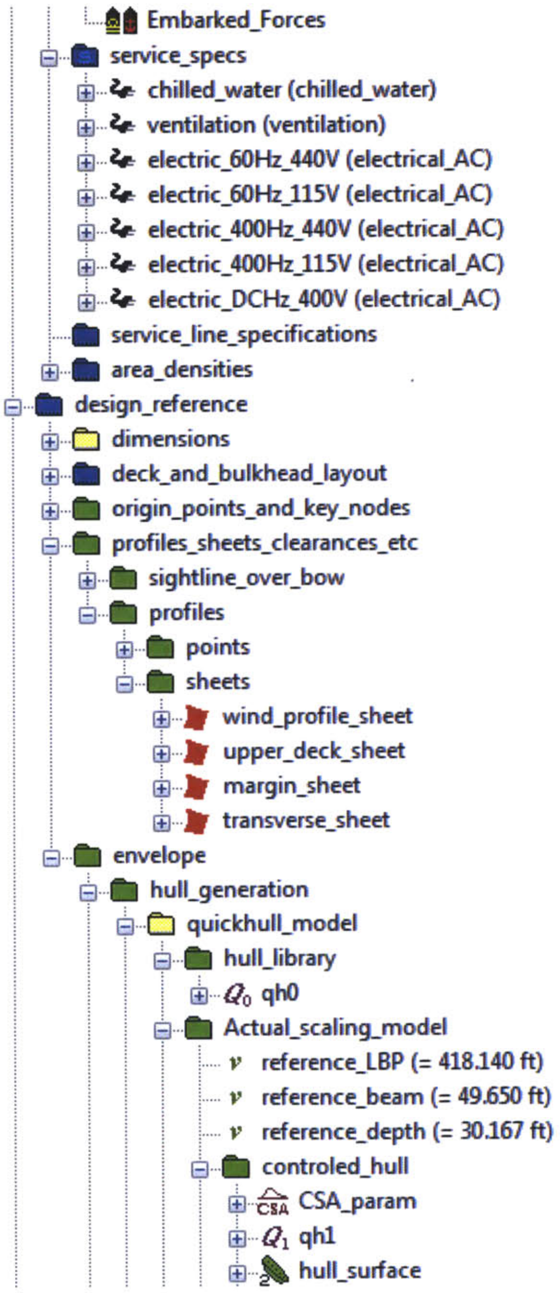
```
fixed_data
Concept
user_variables
    LBP (= 543.293 ft)
    HOA (= 201.015 ft)
    Depth_midships (= 48.015 ft)
    Depth_scaled_from_parent_hull (= 39.197 ft)
    Beam (= 64.528 ft)
    Draft (= 20.464 ft)
    Range (= 5000.000 nm)
    Endurance_Speed (= 18.000 kt)
    Max_Speed (= 30.000 kt)
    prismatic_coefficient (= 0.583)
    midship_coefficient (= 0.775)
    frame_spacing (= 0.000 ft)
    daughter_parent_hull_scaling (= 1.299)
    deck_heights
    bulkheads
setup
    Equipment_Library
        Machinery
            GE_LM6000PD_E_48_5MW
            LM6000PD_47_3MW_GEN
            GENERATOR_LUBE_SYSTEM_SMALL_L
            ENGINE_FUEL_SYSTEM_SMALL_L
            START_AIR_SYSTEM_UNIVERSAL_32_L
            BUS_SWITCHGEAR_LARGE_L
            ENGINE_LUBE_SYSTEM_UNIVERSAL_32_L
            SW_COOL_SYSTEM_LARGE_L
            FIRE_SYSTEM_UNIVERSAL_14_T
            PCM4_5000_T
            PCM1_2400_L
            PCM2_2400_L
            DDA_501_K34G_3_4MW
            501_K34_3MW_GEN
            BUS_SWITCHGEAR_MED_L
            SW_COOL_SYSTEM_SMALL_L
            FIRE_SYSTEM_UNIVERSAL_14_L
```



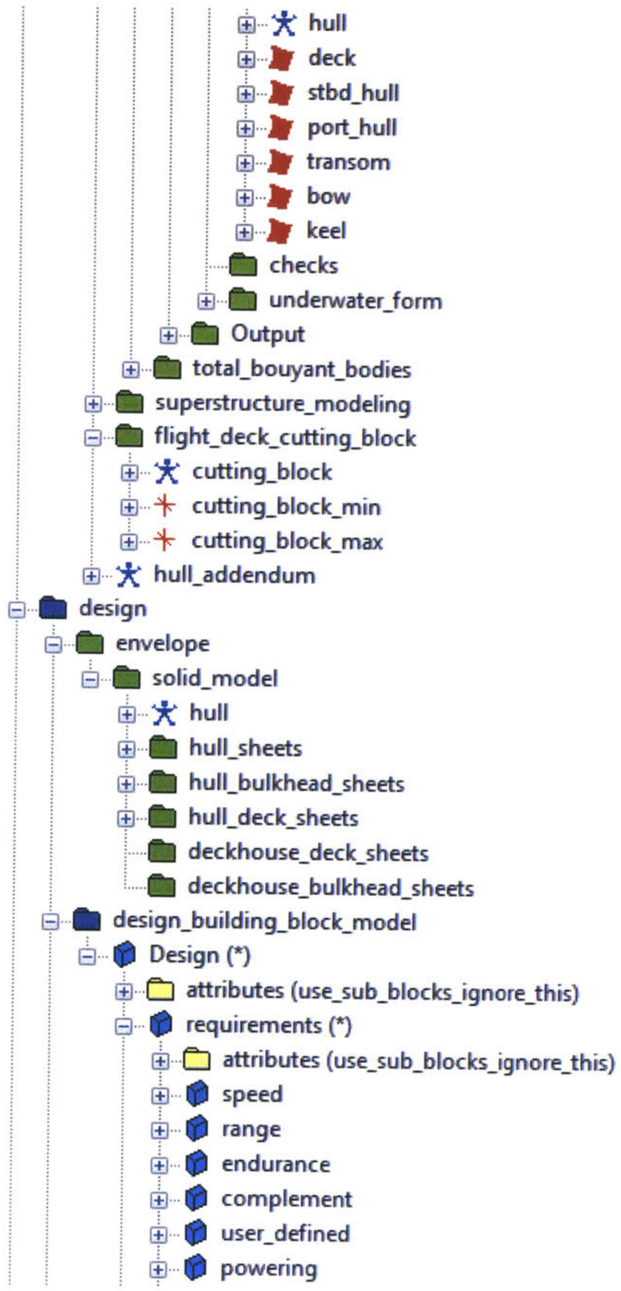
```
      PCM4_5000_L
      POD_SYN_10S_23A_9_8MW_MOTOR
      PCM_POD_SYN_10S_23A_9_11_5MW_L
      PMM_Power_Filter_445_T
      PMM_MOTOR_LUBE_SYSTEM_UNIVERSAL_T
      BRAKING_RESISTOR_069_L
      PCM2_2400_T
      MIL_AIM_28S_15A_28MW_MOTOR
      PCM_MIL_AIM_28S_15A_29_3MW_L
      PMM_Power_Filter_460_T
      BRAKING_RESISTOR_137_L
      DDA_501_K34G_3_4MW_INTAKE_SS_AMR1
      GE_LM6000PD_E_48_5MW_INTAKE_PD_MMR1
      DDA_501_K34G_3_4MW_INTAKE_SS_OMR1
      MIL_AIM_28S_15A_28MW_MOTOR_SHAFT_MMR2
      MIL_AIM_28S_15A_28MW_MOTOR_SHAFT_MMR2_PROP
    Gun
      5_in_54_caliber_Mark_45_gun
      Phalanx_Mk_15_CIWS
    VLS
      Mk_41_64_rds_strike
      Mk_41_128_rds_strike
    Radar
      SPY_1D
    Sonar
      SQS_53C
    Aviation
      flight_deck
      hanger
    Stacks
      stack_1
      stack_2
      stack_3
      stack_4
  ship_conditions
    lightship_condition
    min_op_condition
```



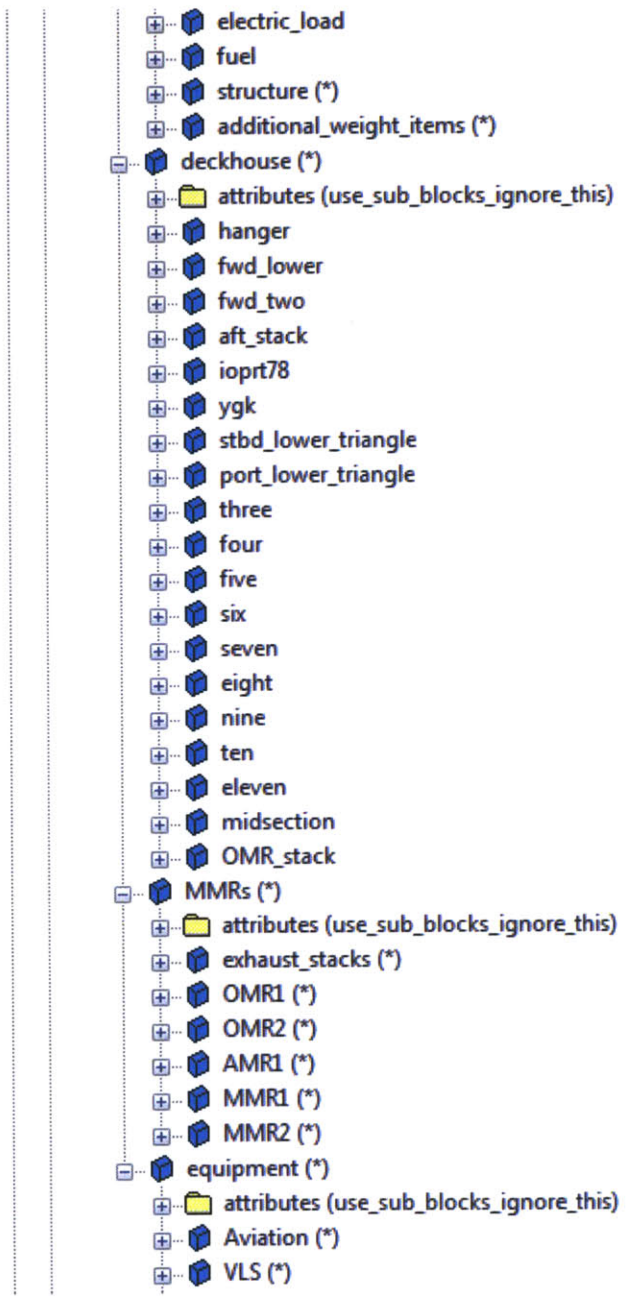
```
full_load_condition
consumable_densities
    sea_water (= 34.938 ft3/LT)
    dieso_fuel (= 40.774 ft3/LT)
    fresh_water (= 35.881 ft3/LT)
    lube_oil (= 39.868 ft3/LT)
tank_levels
    full_tanks (= 0.000)
    low_tanks (= 0.000)
classification_systems
    Group_1_Structures
    Group_2_Propulsion
    Group_3_Electrical
    Group_4_Control_&_Communication
    Group_5_Auxiliary_systems
    Group_6_Outfit_&_Furnishings
    Group_7_Armament
    Group_8_Variable_Load
    unallocated_weight
    machinery_spaces
margins
    overall_weight_margins
    design_margin_multipliers
blocks_to_copy_from
    standard_blocks
        generic_blank_block_type_1
        generic_blank_block_type_2
        generic_blank_block_type_3
user_defined
    Propulsion_Power (units: )
    store_space (units: )
crew_types
    Captain
    XO
    Officers
    CPOs
    Enlisted
```



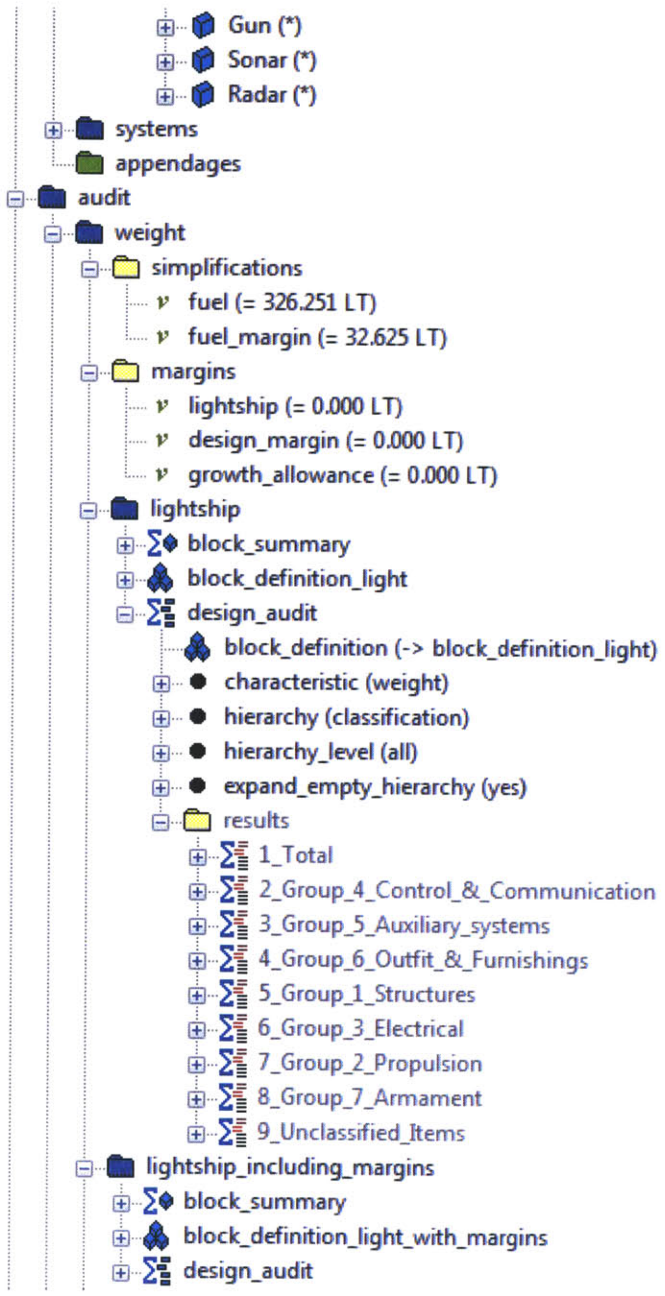
```
      Embarked_Forces
  service_specs
    chilled_water (chilled_water)
    ventilation (ventilation)
    electric_60Hz_440V (electrical_AC)
    electric_60Hz_115V (electrical_AC)
    electric_400Hz_440V (electrical_AC)
    electric_400Hz_115V (electrical_AC)
    electric_DCHz_400V (electrical_AC)
  service_line_specifications
  area_densities
design_reference
  dimensions
  deck_and_bulkhead_layout
  origin_points_and_key_nodes
  profiles_sheets_clearances_etc
    sightline_over_bow
    profiles
      points
      sheets
        wind_profile_sheet
        upper_deck_sheet
        margin_sheet
        transverse_sheet
  envelope
    hull_generation
      quickhull_model
        hull_library
          qh0
        Actual_scaling_model
          reference_LBP (= 418.140 ft)
          reference_beam (= 49.650 ft)
          reference_depth (= 30.167 ft)
          controled_hull
            CSA_param
            qh1
            hull_surface
```



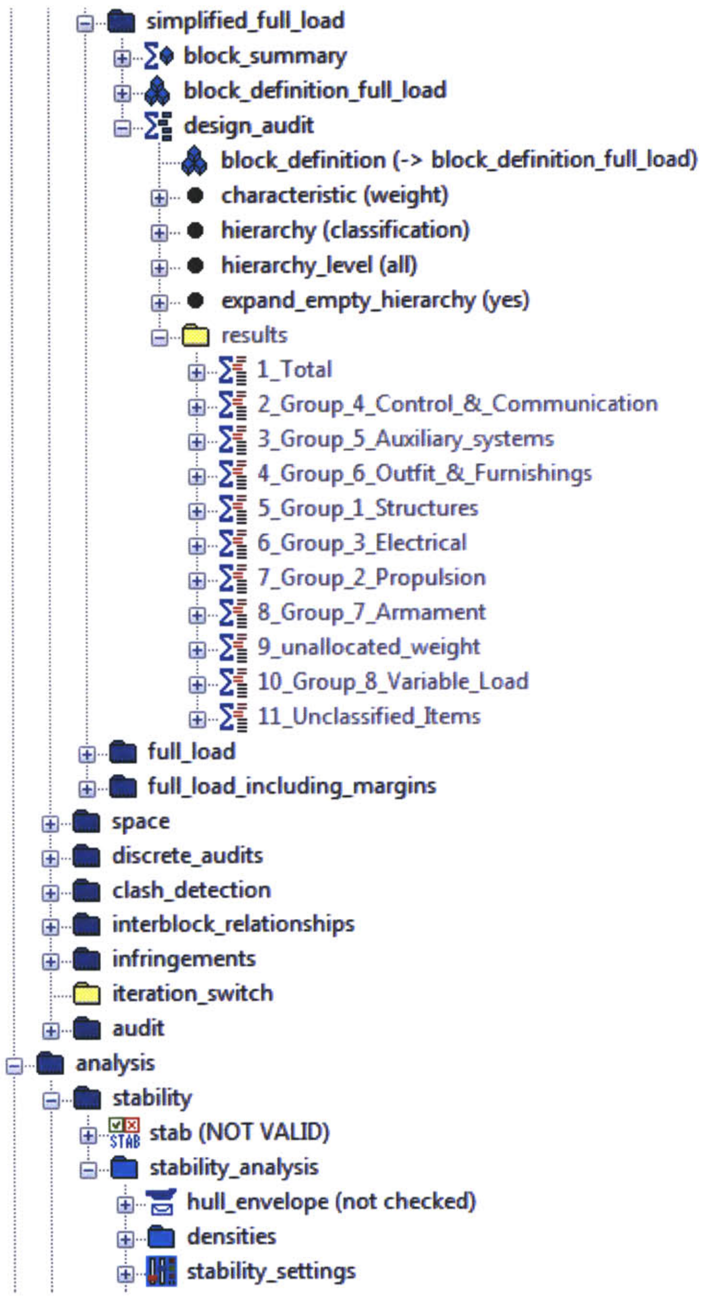
```
                  hull
                  deck
                  stbd_hull
                  port_hull
                  transom
                  bow
                  keel
               checks
               underwater_form
            Output
         total_bouyant_bodies
      superstructure_modeling
      flight_deck_cutting_block
         cutting_block
         cutting_block_min
         cutting_block_max
      hull_addendum
design
   envelope
      solid_model
         hull
         hull_sheets
         hull_bulkhead_sheets
         hull_deck_sheets
         deckhouse_deck_sheets
         deckhouse_bulkhead_sheets
   design_building_block_model
      Design (*)
         attributes (use_sub_blocks_ignore_this)
         requirements (*)
            attributes (use_sub_blocks_ignore_this)
            speed
            range
            endurance
            complement
            user_defined
            powering
```



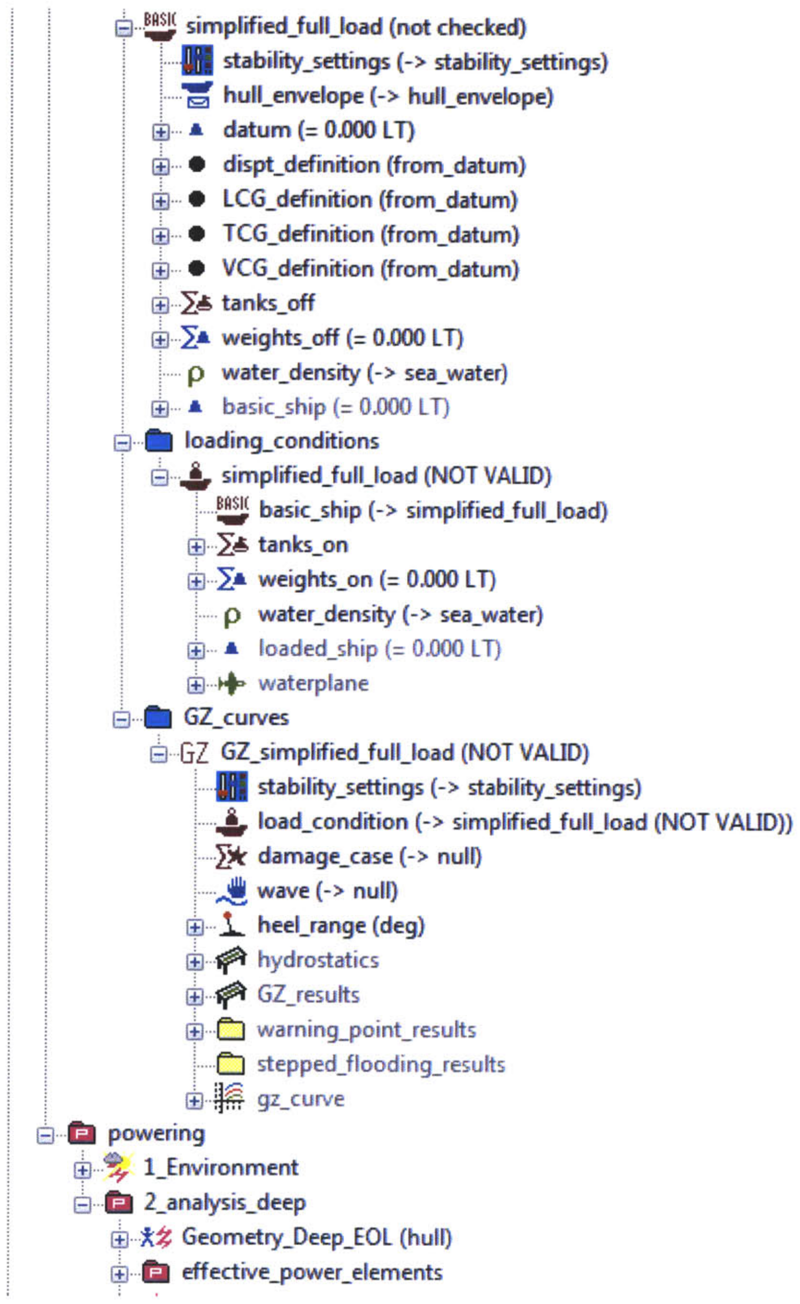
```
    ⊞ electric_load
    ⊞ fuel
    ⊞ structure (*)
    ⊞ additional_weight_items (*)
⊟ deckhouse (*)
    ⊞ attributes (use_sub_blocks_ignore_this)
    ⊞ hanger
    ⊞ fwd_lower
    ⊞ fwd_two
    ⊞ aft_stack
    ⊞ ioprt78
    ⊞ ygk
    ⊞ stbd_lower_triangle
    ⊞ port_lower_triangle
    ⊞ three
    ⊞ four
    ⊞ five
    ⊞ six
    ⊞ seven
    ⊞ eight
    ⊞ nine
    ⊞ ten
    ⊞ eleven
    ⊞ midsection
    ⊞ OMR_stack
⊟ MMRs (*)
    ⊞ attributes (use_sub_blocks_ignore_this)
    ⊞ exhaust_stacks (*)
    ⊞ OMR1 (*)
    ⊞ OMR2 (*)
    ⊞ AMR1 (*)
    ⊞ MMR1 (*)
    ⊞ MMR2 (*)
⊟ equipment (*)
    ⊞ attributes (use_sub_blocks_ignore_this)
    ⊞ Aviation (*)
    ⊞ VLS (*)
```



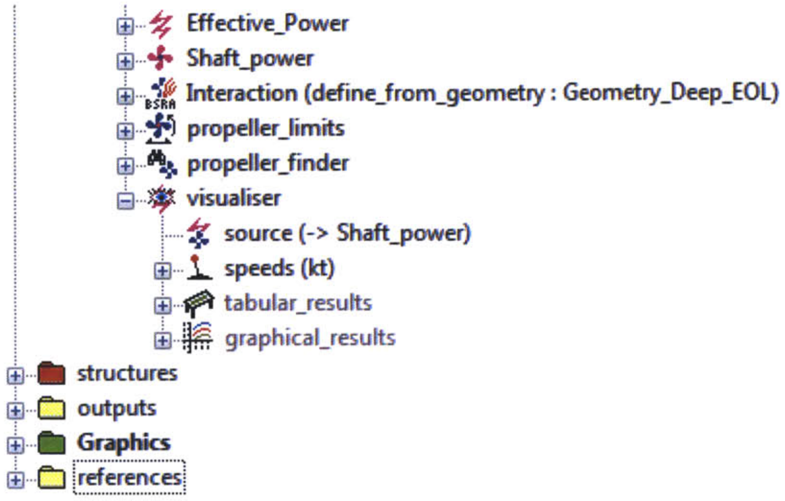
```
            Gun (*)
            Sonar (*)
            Radar (*)
    systems
    appendages
audit
    weight
        simplifications
            fuel (= 326.251 LT)
            fuel_margin (= 32.625 LT)
        margins
            lightship (= 0.000 LT)
            design_margin (= 0.000 LT)
            growth_allowance (= 0.000 LT)
        lightship
            block_summary
            block_definition_light
            design_audit
                block_definition (-> block_definition_light)
                characteristic (weight)
                hierarchy (classification)
                hierarchy_level (all)
                expand_empty_hierarchy (yes)
                results
                    1_Total
                    2_Group_4_Control_&_Communication
                    3_Group_5_Auxiliary_systems
                    4_Group_6_Outfit_&_Furnishings
                    5_Group_1_Structures
                    6_Group_3_Electrical
                    7_Group_2_Propulsion
                    8_Group_7_Armament
                    9_Unclassified_Items
        lightship_including_margins
            block_summary
            block_definition_light_with_margins
            design_audit
```



```
simplified_full_load
    block_summary
    block_definition_full_load
    design_audit
        block_definition (-> block_definition_full_load)
        characteristic (weight)
        hierarchy (classification)
        hierarchy_level (all)
        expand_empty_hierarchy (yes)
        results
            1_Total
            2_Group_4_Control_&_Communication
            3_Group_5_Auxiliary_systems
            4_Group_6_Outfit_&_Furnishings
            5_Group_1_Structures
            6_Group_3_Electrical
            7_Group_2_Propulsion
            8_Group_7_Armament
            9_unallocated_weight
            10_Group_8_Variable_Load
            11_Unclassified_Items
full_load
full_load_including_margins
space
discrete_audits
clash_detection
interblock_relationships
infringements
iteration_switch
audit
analysis
    stability
        stab (NOT VALID)
        stability_analysis
            hull_envelope (not checked)
            densities
            stability_settings
```



```
simplified_full_load (not checked)
    stability_settings (-> stability_settings)
    hull_envelope (-> hull_envelope)
    datum (= 0.000 LT)
    dispt_definition (from_datum)
    LCG_definition (from_datum)
    TCG_definition (from_datum)
    VCG_definition (from_datum)
    tanks_off
    weights_off (= 0.000 LT)
    water_density (-> sea_water)
    basic_ship (= 0.000 LT)
loading_conditions
    simplified_full_load (NOT VALID)
        basic_ship (-> simplified_full_load)
        tanks_on
        weights_on (= 0.000 LT)
        water_density (-> sea_water)
        loaded_ship (= 0.000 LT)
        waterplane
GZ_curves
    GZ_simplified_full_load (NOT VALID)
        stability_settings (-> stability_settings)
        load_condition (-> simplified_full_load (NOT VALID))
        damage_case (-> null)
        wave (-> null)
        heel_range (deg)
        hydrostatics
        GZ_results
        warning_point_results
        stepped_flooding_results
        gz_curve
powering
    1_Environment
    2_analysis_deep
        Geometry_Deep_EOL (hull)
        effective_power_elements
```



Effective_Power
Shaft_power
Interaction (define_from_geometry : Geometry_Deep_EOL)
propeller_limits
propeller_finder
visualiser
source (-> Shaft_power)
speeds (kt)
tabular_results
graphical_results
structures
outputs
Graphics
references

Appendix D: Input for Case Study 1, Mimicking DDG-51

Initial Inputs

INPUTS	
max speed (knots)	29
range (nm)	4400
Installed Power (MW)	108.5_L
Aviation Facilities	yes
Type of Helicoptor Hanger	SH-60F_Seahawk
# of helicoptor hangers	2
hanger orientation	side-by-side
Est Displacement (LT)	9875
Crew Size	323
Endurance (days)	60
Cruising Speed (knots)	20
# of Decks Above MMRs	1
Hull Type	5415_rev2

	INPUTS							
			location					
ITEM #	type	selection	vertical reference	FORE/AFT (% aft from FP)	PORT/STBD (% PORT from CL)	Additional Input Needed	input (ft)	Vertical Adjustment (ft)
1	MMR_Bounding_Boxes	Master Machinery Bounding Box	Atop_Inner_Bottom	47.00%	0.00%	none		
2	Flight_Deck	SH-60F_Seahawk	Weather_Deck	88.00%	0.00%	none		-8.5
3	MMR_Bounding_Boxes	OMR1	Weather_Deck	82.00%	0.00%	none		-29
4			Atop_Inner_Bottom	95.00%	0.00%	none		16
5			Weather_Deck	78.00%	0.00%	none		
6	VLS	Mk_41_32_rds_strike	Weather_Deck	19.00%	0.00%	none		
7	VLS	Mk_41_64_rds_strike	Weather_Deck	75.00%	0.00%	none		9
8	Gun	5_in_54_caliber_Mark_45_gun	Weather_Deck	11.00%	0.00%	Height_of_Hoist	25.5	6
9	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	25.00%	0.00%	none	0	18
10	Gun	Phalanx_Mk_15_CIWS	Weather_Deck	65.00%	0.00%	none	0	9
11	Sonar	SQS_53C	Bottom_of_Hull	2.00%	0.00%	none		
12	Radar	SPY_1D	Weather_Deck	28.50%	60.00%	none		27
13	Radar	SPY_1D	Weather_Deck	28.50%	-60.00%	none		27
14	Radar	SPY_1D	Weather_Deck	34.50%	60.00%	none		36
15	Radar	SPY_1D	Weather_Deck	34.50%	-60.00%	none		36

	bulkhead locations % aft from FP	bulkhead locations ft from FP	bulkhead name
1	86.50%	-415.9379	BH5
2	93.00%	-447.1933	BH6
3	0.00%	0	BH1
4	7.50%	-36.06398	BH2
5	14.50%	-69.72369	BH3
6	77.50%	-372.6611	BH4

Deckhouse Input

Hanger Input	
long. location	76.00%
trans. Location	0.00%
deck	Weather_Deck
vertical disp.	-8.5
dist. btwn. hang.	18
within dkhs?	yes

block #1	
Shape	triangle_type_1
	length
side	(ft)
1	21.21320344
2	15
3	15
Bottom Level	Weather_Deck
Top Level	01
Long. Location (ft from FP)	125
Trans Location (ft to PORT)	17.5
Name	port_fwd

block #2	
Shape	triangle_type_2
	length
side	(ft)
1	21.21320344
2	15
3	15
Bottom Level	Weather_Deck
Top Level	01
Long. Location (ft from FP)	125
Trans Location (ft to PORT)	-17.5
Name	stbd_fwd

block #3	
Shape	rectangle
	length
side	(ft)
1	35
2	15
3	35
4	15
Bottom Level	Weather_Deck
Top Level	O1
Long. Location (ft from FP)	117.5
Trans Location (ft to PORT)	0
Name	cent_fwd_lower

block #4	
Shape	rectangle
	length
side	(ft)
1	65
2	75
3	65
4	75
Bottom Level	Weather_Deck
Top Level	O1
Long. Location (ft from FP)	162.5
Trans Location (ft to PORT)	0
Name	center_bottom

block #5	
Shape	triangle_type_3
	length
side	(ft)
1	37.12310601
2	26.25
3	26.25
Bottom Level	Weather_Deck
Top Level	O1
Long. Location (ft from FP)	200
Trans Location (ft to PORT)	-6.25
Name	stbd_lower_aft

block #6	
Shape	triangle_type_4
	length
side	(ft)
1	37.12310601
2	26.25
3	26.25
Bottom Level	Weather_Deck
Top Level	01
Long. Location (ft from FP)	200
Trans Location (ft to PORT)	6.25
Name	port_lower_aft

block #7	
Shape	rectangle
	length
side	(ft)
1	12.5
2	26.25
3	12.5
4	26.25
Bottom Level	Weather_Deck
Top Level	01
Long. Location (ft from FP)	213.125
Trans Location (ft to PORT)	0
Name	aft_lower_center

block #8	
Shape	rectangle
	length
side	(ft)
1	60
2	69
3	60
4	69
Bottom Level	Weather_Deck
Top Level	01
Long. Location (ft from FP)	362.5
Trans Location (ft to PORT)	0
Name	hangar

block #9	
Shape	rectangle
	length
side	(ft)
1	35
2	88
3	35
4	88
Bottom Level	Weather_Deck
Top Level	O1
Long. Location (ft from FP)	284
Trans Location (ft to PORT)	5
Name	fwd_of_hangar
block #10	
Shape	rectangle
	length
side	(ft)
1	21.5
2	58.5
3	21.5
4	58.5
Bottom Level	O1
Top Level	O5
Long. Location (ft from FP)	271
Trans Location (ft to PORT)	0
Name	aft_stack
block #11	
Shape	rectangle
	length
side	(ft)
1	61
2	54.5
3	61
4	54.5
Bottom Level	O1
Top Level	O2
Long. Location (ft from FP)	167
Trans Location (ft to PORT)	0
Name	midlevel_center

block #12	
Shape	triangle_type_1
	length
side	(ft)
1	18.38477631
2	13
3	13
Bottom Level	O1
Top Level	O2
Long. Location (ft from FP)	139.75
Trans Location (ft to PORT)	17.5
Name	center_fwd_port

block #13	
Shape	triangle_type_2
	length
side	(ft)
1	18.38477631
2	13
3	13
Bottom Level	O1
Top Level	O2
Long. Location (ft from FP)	139.75
Trans Location (ft to PORT)	-17.5
Name	center_fwd_stbd

block #14	
Shape	rectangle
	length
side	(ft)
1	35
2	26
3	35
4	26
Bottom Level	O1
Top Level	O2
Long. Location (ft from FP)	126.75
Trans Location (ft to PORT)	0
Name	center_fwd

block #15	
Shape	rectangle
	length
side	(ft)
1	21.5
2	26
3	21.5
4	26
Bottom Level	O1
Top Level	O5
Long. Location (ft from FP)	207
Trans Location (ft to PORT)	0
Name	fwd_stack

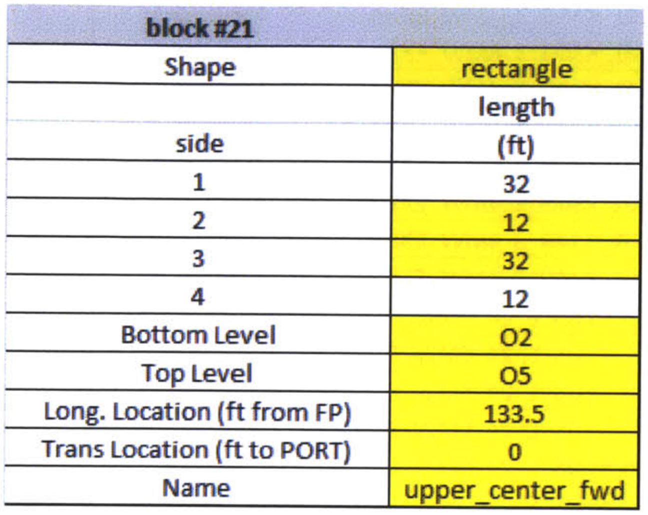
block #16	
Shape	rectangle
	length
side	(ft)
1	57
2	23
3	57
4	23
Bottom Level	O2
Top Level	O5
Long. Location (ft from FP)	151
Trans Location (ft to PORT)	0
Name	top_center

block #17	
Shape	triangle_type_1
	length
side	(ft)
1	16.97056275
2	12
3	12
Bottom Level	O2
Top Level	O5
Long. Location (ft from FP)	139.5
Trans Location (ft to PORT)	16
Name	upper_fwd_port

block #18	
Shape	triangle_type_2
	length
side	(ft)
1	16.97056275
2	12
3	12
Bottom Level	O2
Top Level	O5
Long. Location (ft from FP)	139.5
Trans Location (ft to PORT)	-16
Name	upper_fwd_stbd

block #19	
Shape	triangle_type_3
	length
side	(ft)
1	40.30508653
2	28.5
3	28.5
Bottom Level	O2
Top Level	O5
Long. Location (ft from FP)	162.5
Trans Location (ft to PORT)	0
Name	upper_aft_stbd

block #20	
Shape	triangle_type_4
	length
side	(ft)
1	40.30508653
2	28.5
3	28.5
Bottom Level	O2
Top Level	O5
Long. Location (ft from FP)	162.5
Trans Location (ft to PORT)	0
Name	upper_aft_port



block #21	
Shape	rectangle
	length
side	(ft)
1	32
2	12
3	32
4	12
Bottom Level	O2
Top Level	O5
Long. Location (ft from FP)	133.5
Trans Location (ft to PORT)	0
Name	upper_center_fwd

Stack Adjustment

Point #	Exhaust	User Adjustment (ft)		
		x	y	z
1	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_1	0	0	0
2	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_2	0	0	-20
3	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_3	-20	0	-10
4	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_4	-40	0	0
5	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_5	-40	0	0
6	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_6	-40	0	0
7	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_7	-40	0	0
8	DDA_501_K34G_3_4MW_EXHAUST_SS_AMR1_p_8	-40	0	0
1	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_1	0	0	0
2	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_2	0	0	0
3	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_3	-35	0	0
4	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_4	-35	-1	0
5	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_5	-35	-2	0
6	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_6	-35	-3	0
7	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_7	-35	-4	0
8	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_8	-35	-5	0
1	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_1	0	0	0
2	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_2	0	0	0
3	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_3	0	0	0
4	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_4	-35	1	0
5	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_5	-35	2	0
6	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_6	-35	3	0
7	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_7	-35	4	0
8	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR1_p_8	-35	5	0
1	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_1	0	0	0
2	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_2	0	0	-10
3	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_3	35	-8	-10
4	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_4	54	-8	0
5	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_5	56	0	0
6	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_6	60	0	0
7	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_7	60	0	0
8	DDA_501_K34G_3_4MW_EXHAUST_SS_MMR4_p_8	60	0	0
1	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_1	0	0	0
2	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_2	0	4	-10
3	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_3	66	0	-10
4	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_4	66	0	0
5	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_5	66	-4	0
6	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_6	66	-4	0
7	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_7	66	-4	0
8	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_8	66	-4	0

1	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_1	0	0	0
2	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_2	0	12	-10
3	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_3	40	12	-10
4	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_4	61	5	0
5	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_5	61	5	0
6	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_6	61	4	0
7	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_7	61	4	0
8	GE_LM2500_PLUS_E_24_6MW_EXHAUST_PD_MMR4_p_8	61	4	0
1	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_1	0	0	0
2	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_2	0	0	0
3	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_3	0	0	0
4	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_4	0	0	0
5	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_5	0	0	0
6	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_6	0	0	0
7	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_7	0	0	0
8	OMR1_DDA_501_K34G_3_4MW_EXHAUST_SS_OMR1_p_8	0	0	0

(THIS PAGE INTENTIONALLY LEFT BLANK)

Appendix E: ESSDT Code

Code for Suppressing Calculations within Excel

```
Sub Supress_Calculations()
' Supress_Calculations Macro

   Application.Calculation = xlManual
End Sub
Sub Update_Calculations()

' Update_Calculations Macro

   Application.Calculation = xlAutomatic
   Application.Calculation = xlManual

   Worksheets("Power_Bank").Activate
   Set Power_Needed = Range("G11")
   Set Power_Available = Range("C16")
   Deficit = Power_Needed - Power_Available
   Worksheets("Initial_Input").Activate
   If Power_Needed > Power_Available Then

      MsgBox "WARNING! Potential maximum power needed exceeds power provided by " & Deficit &
"MW"

   End If

End Sub
Sub Enable_Calculations()

' Enable_Calculations Macro

   Application.Calculation = xlAutomatic
End Sub

Sub Update_Deckhouse()

' Update_Calculations Macro

   Application.Calculation = xlAutomatic
   Application.Calculation = xlManual

   Worksheets("Power_Bank").Activate
   Set Power_Needed = Range("G11")
   Set Power_Available = Range("C16")
   Deficit = Power_Needed - Power_Available
   Worksheets("Deckhouse_Input").Activate
   If Power_Needed > Power_Available Then
```

```
      MsgBox "WARNING! Potential maximum power needed exceeds power provided by " & Deficit &
"MW"

    End If

End Sub

Sub Update_Buttons()

' Update_Calculations Macro

    Application.Calculation = xlAutomatic
    Application.Calculation = xlManual

    Worksheets("Power_Bank").Activate
    Set Power_Needed = Range("G11")
    Set Power_Available = Range("C16")
    Deficit = Power_Needed - Power_Available
    Worksheets("Import_to_Paramarine").Activate
    If Power_Needed > Power_Available Then

      MsgBox "WARNING! Potential maximum power needed exceeds power provided by " & Deficit &
"MW"

    End If

End Sub

Sub Update_Stacks()

' Update_Calculations Macro

    Application.Calculation = xlAutomatic
    Application.Calculation = xlManual

End Sub
```

Code for Creating File Structure Within Paramarine

```
Sub create_file_structure()

  Call find_files
    'find_files is in Generic Functions
    Call open_output_file(output_file)

      Print #1, "closedesign"
      Print #1, "newdesign"
      Print #1, "{fixed_data.units.length_default.ft}"
      Print #1, "{fixed_data.units.mass_default.LT}"
      Print #1, "{fixed_data.units.volume_default.ft3}"

  Print #1, "playback C:\ESSDT\Macros\Early_stage_design.kcl"

  Worksheets("Defaults").Activate


''''''''''''''''''''
'Insert Primary Files'
''''''''''''''''''''


Print #1, "new placeholder user_variables"
Print #1, "new concept_placeholder setup"
Print #1, "new concept_placeholder design_reference"
Print #1, "new concept_placeholder design"
Print #1, "new concept_placeholder audit"
Print #1, "new concept_placeholder analysis"
Print #1, "new struct_placeholder structures"
Print #1, "new placeholder outputs"
Print #1, "new geom_placeholder Graphics"
Print #1, "new placeholder references"
  Print #1, "{references} new geom_placeholder origin"
    Print #1, "{references.origin} new point origin"
  Print #1, "{references} new geom_placeholder deck_points"
  Print #1, "{references} new geom_placeholder bulkhead_points"

''''''''''''''''''''
'Insert user variables'
''''''''''''''''''''


  Print #1, "{user_variables} new var LBP"
  Print #1, "{user_variables.LBP} set_units_category (length_default)"
  Print #1, "{user_variables} new var HOA"
  Print #1, "{user_variables.HOA} set_units_category (length_default)"
  Print #1, "{user_variables} new var Depth_midships"
  Print #1, "{user_variables.Depth_midships} set_units_category (length_default)"
  Print #1, "{user_variables} new var Depth_scaled_from_parent_hull"
  Print #1, "{user_variables.Depth_scaled_from_parent_hull} set_units_category (length_default)"
  Print #1, "{user_variables} new var Beam"
  Print #1, "{user_variables.Beam} set_units_category (length_default)"
  Print #1, "{user_variables} new var Draft"
  Print #1, "{user_variables.Draft} set_units_category (length_default)"
' Print #1, "{user_variables} new var Continuous_Decks_above_MMRs"
  Print #1, "{user_variables} new var Range"
  Print #1, "{user_variables.Range} set_units_category (length_major)"
  Print #1, "{user_variables} new var Endurance_Speed"
  Print #1, "{user_variables.Endurance_Speed} set_units_category (linear_velocity)"
  Print #1, "{user_variables} new var Max_Speed"
  Print #1, "{user_variables.Max_Speed} set_units_category (linear_velocity)"
  Print #1, "{user_variables} new var prismatic_coefficient"
  Print #1, "{user_variables} new var midship_coefficient"
  Print #1, "{user_variables} new var frame_spacing"
  Print #1, "{user_variables.frame_spacing} set_units_category (length_default)"
  Print #1, "{user_variables} new var daughter_parent_hull_scaling"
  Print #1, "{user_variables} new placeholder deck_heights"
  Print #1, "{user_variables} new placeholder bulkheads"


''''''''''''''''''''''''''''
'Insert Setup secondary files'
''''''''''''''''''''''''''''

  Print #1, "{setup} new concept_placeholder Equipment_Library"
  Print #1, "{setup} new condition_container ship_conditions"
  Print #1, "{setup} new consumables_container consumable_densities"
  Print #1, "{setup} new concept_placeholder tank_levels"
  Print #1, "{setup} new concept_placeholder classification_systems"
  Print #1, "{setup} new concept_placeholder margins"
  Print #1, "{setup} new concept_placeholder blocks_to_copy_from"
  Print #1, "{setup} new user_spec_container user_defined"
  Print #1, "{setup} new personnel_types_container crew_types"
  Print #1, "{setup} new service_container service_specs"
  Print #1, "{setup} new concept_placeholder service_line_specifications"
  Print #1, "{setup} new concept_placeholder area_densities"

''''''''''''''''''''''''''''''''''''''
'Insert design_reference secondary files'
''''''''''''''''''''''''''''''''''''''

  Print #1, "{design_reference} new placeholder dimensions"
  Print #1, "{design_reference} new concept_placeholder deck_and_bulkhead_layout"
  Print #1, "{design_reference} new geom_placeholder origin_points_and_key_nodes"
  Print #1, "{design_reference} new geom_placeholder profiles_sheets_clearances_etc"
  Print #1, "{design_reference} new geom_placeholder envelope"
```

```
''''''''''''''''''''''''''''''
'Insert design secondary files'
''''''''''''''''''''''''''''''

    Print #1, "{design} new geom_placeholder envelope"
    Print #1, "{design} new concept_placeholder design_building_block_model"
    Print #1, "{design} new concept_placeholder systems"
    Print #1, "{design} new geom_placeholder appendages"

'''''''''''''''''''''''''''''
'Insert audit secondary files'
'''''''''''''''''''''''''''''

    Print #1, "{audit} new concept_placeholder weight"
    Print #1, "{audit} new concept_placeholder space"
    Print #1, "{audit} new concept_placeholder discrete_audits"
    Print #1, "{audit} new concept_placeholder clash_detection"
    Print #1, "{audit} new concept_placeholder interblock_relationships"
    Print #1, "{audit} new concept_placeholder infringements"
    Print #1, "{audit} new placeholder iteration_switch"

''''''''''''''''''''''''''''''''
'Insert analysis secondary files'
''''''''''''''''''''''''''''''''

  '  Print #1, "{analysis} new concept_placeholder esd_intact_stability"
        Print #1, "{analysis.esd_intact_stability} new stab_placeholder 1_criteria"
        Print #1, "{analysis.esd_intact_stability} new stability_analysis 2_stability_without_margins"

''''''''''''''''''''''''''''''''''
'Insert structures secondary files'
''''''''''''''''''''''''''''''''''

    Print #1, "{structures} new placeholder config_controls"
    Print #1, "{structures} new placeholder input_dimentions"
    Print #1, "{structures} new stiffener_profile_holder profiles"
    Print #1, "{structures} new struct_placeholder tolerances"
    Print #1, "{structures} new struct_placeholder materials"
    Print #1, "{structures} new struct_placeholder plate_and_stiffener_specifications"
    Print #1, "{structures} new struct_placeholder stiffener_schemas"
    Print #1, "{structures} new struct_placeholder scantlings"
    Print #1, "{structures} new struct_placeholder panel_generation"
    Print #1, "{structures} new struct_placeholder final_structure"

'''''''''''''''''''''''''''''''
'Insert outputs secondary files'
'''''''''''''''''''''''''''''''
```

```
  Print #1, "{outputs} new placeholder controls"
  Print #1, "{outputs} new placeholder hullform_lines_plan"
  Print #1, "{outputs} new placeholder general_arrangement_drawing"
  Print #1, "{outputs} new excel_workbook outputs"
    Print #1, "{outputs.outputs} new excel_worksheet for_ESSDT"


''''''''''''''''''''''''''
'Insert setup tertiary files'
''''''''''''''''''''''''''

    Print #1, "{setup.Equipment_Library} new equipment_container Machinery"
    Print #1, "{setup.Equipment_Library} new equipment_container Gun"
    Print #1, "{setup.Equipment_Library} new equipment_container VLS"
    Print #1, "{setup.Equipment_Library} new equipment_container Radar"
    Print #1, "{setup.Equipment_Library} new equipment_container Sonar"
'    Print #1, "{setup.Equipment_Library} new equipment_container Podded_Propulsion"
    Print #1, "{setup.Equipment_Library} new equipment_container Aviation"
    Print #1, "{setup.Equipment_Library} new equipment_container Stacks"

    Print #1, "{setup.ship_conditions} new condition lightship_condition"
    Print #1, "{setup.ship_conditions} new condition min_op_condition"
    Print #1, "{setup.ship_conditions} new condition full_load_condition"
    Print #1, "{setup.consumable_densities} new density sea_water"
    Print #1, "{setup.consumable_densities} new density dieso_fuel"
    Print #1, "{setup.consumable_densities} new density ballast =
(setup.consumable_densities.sea_water)"

    Print #1, "{setup.tank_levels} new var full_tanks"
    Print #1, "{setup.tank_levels} new var low_tanks"

    Print #1, "{setup.classification_systems} new classification Group_1_Structures"
    Print #1, "{setup.classification_systems} new classification Group_2_Propulsion"
    Print #1, "{setup.classification_systems} new classification Group_3_Electrical"
    Print #1, "{setup.classification_systems} new classification Group_4_Control_&_Communication"
    Print #1, "{setup.classification_systems} new classification Group_5_Auxiliary_systems"
    Print #1, "{setup.classification_systems} new classification Group_6_Outfit_&_Furnishings"
    Print #1, "{setup.classification_systems} new classification Group_7_Armament"
    Print #1, "{setup.classification_systems} new classification Group_8_Variable_Load"
    Print #1, "{setup.classification_systems} new classification unallocated_weight"
    Print #1, "{setup.classification_systems} new classification machinery_spaces"

    Print #1, "{setup.margins} new placeholder overall_weight_margins"
      Print #1, "{setup.margins.overall_weight_margins} new var growth_allowance"
        Print #1, "{setup.margins.overall_weight_margins.growth_allowance} set_units_category
(proportion)"
'     Print #1, "{setup.margins.overall_weight_margins} new var board_margin"
'        Print #1, "{setup.margins.overall_weight_margins.board_margin} set_units_category
(proportion)"
        Print #1, "{setup.margins.overall_weight_margins} new var overall_design_margin"
           Print #1, "{setup.margins.overall_weight_margins.overall_design_margin} set_units_category
(proportion)"

      Print #1, "{setup.margins} new placeholder design_margin_multipliers"
'         Print #1, "{setup.margins.design_margin_multipliers} new var
group_1_structures_weight_margin"
 '        Print #1, "{setup.margins.design_margin_multipliers} new var
group_1_structures_space_margin"
  '       Print #1, "{setup.margins.design_margin_multipliers} new var
group_2_propulsion_weight_margin"
   '      Print #1, "{setup.margins.design_margin_multipliers} new var
group_2_propulsion_space_margin"
    '     Print #1, "{setup.margins.design_margin_multipliers} new var
group_3_electrical_weight_margin"
     '    Print #1, "{setup.margins.design_margin_multipliers} new var group_3_electrical_space_margin"
      '   Print #1, "{setup.margins.design_margin_multipliers} new var
group_4_control_&_communication_weight_margin"
       '  Print #1, "{setup.margins.design_margin_multipliers} new var
group_4_control_&_communication_space_margin"
        ' Print #1, "{setup.margins.design_margin_multipliers} new var
group_5_auxiliary_systems_weight_margin"
         ' Print #1, "{setup.margins.design_margin_multipliers} new var
group_5_auxiliary_systems_space_margin"
          ' Print #1, "{setup.margins.design_margin_multipliers} new var
group_6_outfit_&_furnishings_weight_margin"
'         Print #1, "{setup.margins.design_margin_multipliers} new var
group_6_outfit_&_furnishins_space_margin"
 '        Print #1, "{setup.margins.design_margin_multipliers} new var
group_7_armament_weight_margin"
  '       Print #1, "{setup.margins.design_margin_multipliers} new var
group_7_armament_space_margin"
   '      Print #1, "{setup.margins.design_margin_multipliers} new var
group_8_variable_load_weight_margin"
    '     Print #1, "{setup.margins.design_margin_multipliers} new var
group_8_variable_load_space_margin"
     '    Print #1, "{setup.margins.design_margin_multipliers} new var passageway_space_margin"
      '      Print #1, "{setup.margins.design_margin_multipliers.group_1_structures_weight_margin}
set_units_category (proportion)"
       '     Print #1, "{setup.margins.design_margin_multipliers.group_1_structures_space_margin}
set_units_category (proportion)"
        '    Print #1, "{setup.margins.design_margin_multipliers.group_2_propulsion_weight_margin}
set_units_category (proportion)"
         '   Print #1, "{setup.margins.design_margin_multipliers.group_2_propulsion_space_margin}
set_units_category (proportion)"
```

```
'   Print #1, "{setup.margins.design_margin_multipliers.group_3_electrical_weight_margin}
set_units_category (proportion)"
'   Print #1, "{setup.margins.design_margin_multipliers.group_3_electrical_space_margin}
set_units_category (proportion)"
'  Print #1,
"{setup.margins.design_margin_multipliers.group_4_control_&_communication_weight_margin}
set_units_category (proportion)"
'  Print #1,
"{setup.margins.design_margin_multipliers.group_4_control_&_communication_space_margin}
set_units_category (proportion)"
' Print #1,
"{setup.margins.design_margin_multipliers.group_5_auxiliary_systems_weight_margin}
set_units_category (proportion)"
' Print #1,
"{setup.margins.design_margin_multipliers.group_5_auxiliary_systems_space_margin}
set_units_category (proportion)"
'Print #1,
"{setup.margins.design_margin_multipliers.group_6_outfit_&_furnishings_weight_margin}
set_units_category (proportion)"
'          Print #1,
"{setup.margins.design_margin_multipliers.group_6_outfit_&_furnishings_space_margin}
set_units_category (proportion)"
'          Print #1, "{setup.margins.design_margin_multipliers.group_7_armament_weight_margin}
set_units_category (proportion)"
'          Print #1, "{setup.margins.design_margin_multipliers.group_7_armament_space_margin}
set_units_category (proportion)"
'          Print #1, "{setup.margins.design_margin_multipliers.group_8_variable_load_weight_margin}
set_units_category (proportion)"
'          Print #1, "{setup.margins.design_margin_multipliers.group_8_variable_load_space_margin}
set_units_category (proportion)"
'          Print #1, "{setup.margins.design_margin_multipliers.passageway_space_margin}
set_units_category (proportion)"

        Print #1, "{setup.margins.design_margin_multipliers} new var weight_margin"
        Print #1, "{setup.margins.design_margin_multipliers} new var power_margin"
        Print #1, "{setup.margins.design_margin_multipliers} new var volume_margin"
        Print #1, "{setup.margins.design_margin_multipliers} new var weight_allowance"
        Print #1, "{setup.margins.design_margin_multipliers} new var KG_allowance"
        Print #1, "{setup.margins.design_margin_multipliers} new var Power_allowance"
          Print #1, "{setup.margins.design_margin_multipliers.weight_margin} set_units_category
(proportion)"
          Print #1, "{setup.margins.design_margin_multipliers.power_margin} set_units_category
(proportion)"
          Print #1, "{setup.margins.design_margin_multipliers.volume_margin} set_units_category
(proportion)"
          Print #1, "{setup.margins.design_margin_multipliers.weight_allowance} set_units_category
(proportion)"
```

```
            Print #1, "{setup.margins.design_margin_multipliers.KG_allowance} set_units_category
(proportion)"
            Print #1, "{setup.margins.design_margin_multipliers.Power_allowance} set_units_category
(proportion)"

            Set margins = Range("C38:C45")

            Print #1, "{setup.margins.design_margin_multipliers.weight_margin} = " & margins(3)
            Print #1, "{setup.margins.design_margin_multipliers.power_margin} = " & margins(1)
            Print #1, "{setup.margins.design_margin_multipliers.volume_margin} = " & margins(2)
            Print #1, "{setup.margins.design_margin_multipliers.weight_allowance} = " & margins(6)
            Print #1, "{setup.margins.design_margin_multipliers.KG_allowance} = " & margins(7)
            Print #1, "{setup.margins.design_margin_multipliers.Power_allowance} = " & margins(8)

        Print #1, "{setup.blocks_to_copy_from} new concept_placeholder standard_blocks"
          Print #1, "{setup.blocks_to_copy_from.standard_blocks} new building_block
generic_blank_block_type_1"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var weight_margin"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var space_margin"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var X_fwd"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_
fwd} set_units_category (length_default)"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var X_aft"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_
aft} set_units_category (length_default)"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var X_size"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_
size} set_units_category (length_default)"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_
size} =
(setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_f
wd-
setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_a
ft)"
```

```
        Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var Y_port"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_
port} set_units_category (length_default)"
        Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var Y_stbd"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_
stbd} set_units_category (length_default)"
        Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var Y_size"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_
size} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_
size} =
(setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_
port-
setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_st
bd)"
        Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var Z_top"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_
top} set_units_category (length_default)"
        Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var Z_btm"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_
btm} set_units_category (length_default)"
        Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var Z_size"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_
size} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_
size} =
(setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_t
op-
setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_b
tm)"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var design_gross_volume_"
               Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.de
sign_gross_volume_} set_units_category (volume_default)"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new var design_nett_volume"
               Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.de
sign_net_volume} set_units_category (volume_default)"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new char_space area"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics}
new char_weight System_weight"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.solid}
cuboid_from_extents
(setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_s
ize,
setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_si
ze,
setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_si
ze )"
            'Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.solid} translate
((setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.X_
fwd+setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristic
s.X_aft)/2,
(setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Y_
port+setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristic
s.Y_stbd)/2,
(setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.Z_t
op+setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1.attributes.characteristics.
Z_btm)/2 )"


        Print #1, "{setup.blocks_to_copy_from.standard_blocks} new building_block
generic_blank_block_type_2"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var weight_margin"
```

```
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var space_margin"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var X_size"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics.X_
size} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var Y_size"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics.Y_
size} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var Z_size"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics.Z_
size} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var deck"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics.de
ck} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var design_gross_volume_"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics.de
sign_gross_volume_} set_units_category (volume_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new var design_nett_volume"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics.de
sign_net_volume} set_units_category (volume_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new char_space area"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_2.attributes.characteristics}
new char_weight System_weight"
```

```
        Print #1, "{setup.blocks_to_copy_from.standard_blocks} new building_block
generic_blank_block_type_3"
            Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new var weight_margin"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new var space_margin"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new var X_pos"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics.X_
pos} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new var Y_pos"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics.Y_
pos} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new var Z_pos"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics.Z_
pos} set_units_category (length_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new var design_gross_volume_"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics.de
sign_gross_volume_} set_units_category (volume_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new var design_nett_volume"
              Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics.de
sign_net_volume} set_units_category (volume_default)"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new char_space area"
          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_3.attributes.characteristics}
new char_weight System_weight"

'       Print #1, "{setup.blocks_to_copy_from.standard_blocks} new building_block
generic_blank_block_type_4"
```

```
'          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var weight_margin"
 '          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var space_margin"
  '         Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var X_fwd"
   '          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_4_side.attributes.characteristics.X_
fwd} set_units_category (length_default)"
    '       Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var X_aft"
     '          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_4_side.attributes.characteristics.X_
aft} set_units_category (length_default)"
      '     Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var X_fwd"
       '          Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_4_side.attributes.characteristics.X_
fwd} set_units_category (length_default)"
        '    Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var X_fwd"
         '       Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_4_side.attributes.characteristics.X_
fwd} set_units_category (length_default)"
          '  Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var X_fwd"
           '      Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_4_side.attributes.characteristics.X_
fwd} set_units_category (length_default)"
            ' Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_4.attributes.characteristics}
new var X_fwd"
             '    Print #1,
"{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_4_side.attributes.characteristics.X_
fwd} set_units_category (length_default)"

    Print #1, "{setup.user_defined} new user_specification Propulsion_Power"
    Print #1, "{setup.user_defined} new user_specification store_space"

    Print #1, "{setup.crew_types} new personnel_type Captain"
    Print #1, "{setup.crew_types} new personnel_type XO"
```

```
        Print #1, "{setup.crew_types} new personnel_type Officers"
        Print #1, "{setup.crew_types} new personnel_type CPOs"
        Print #1, "{setup.crew_types} new personnel_type Enlisted"
        Print #1, "{setup.crew_types} new personnel_type Embarked_Forces"

        'Fill Service Spec folder from main programs

        Print #1, "{setup.service_specs} new service_specification chilled_water"
          Print #1, "{setup.service_specs.chilled_water.chilled_water}"
        Print #1, "{setup.service_specs} new service_specification ventilation"
          Print #1, "{setup.service_specs.ventilation.ventilation}"

        'Finish filling service line spec folder from main programs

        Print #1, "{setup.area_densities} new var ships_stores"
          Print #1, "{setup.area_densities.ships_stores} set_units_category (mass_per_area_default)"

''''''''''''''''''''''''''''''''''''''''
'Insert Design Reference tertiary files'
''''''''''''''''''''''''''''''''''''''''

        'dimensions
        Print #1, "{design_reference.dimensions} new placeholder superstructure_dimensions"
          Print #1, "{design_reference.dimensions.superstructure_dimensions} new var angle_on_sides"
            Print #1, "{design_reference.dimensions.superstructure_dimensions.angle_on_sides}
set_units_category (angle)"
          Print #1, "{design_reference.dimensions.superstructure_dimensions} new var angle_on_ends"
            Print #1, "{design_reference.dimensions.superstructure_dimensions.angle_on_ends}
set_units_category (angle)"
        Print #1, "{design_reference.dimensions} new placeholder other_dimensions"
          Print #1, "{design_reference.dimensions.other_dimensions} new var shaft_depression"
            Print #1, "{design_reference.dimensions.other_dimensions.shaft_depression}
set_units_category (angle)"

        'deck & bulkhead layout
        Print #1, "{design_reference.deck_and_bulkhead_layout} new placeholder deck_height"
        Print #1, "{design_reference.deck_and_bulkhead_layout} new placeholder bulkheads"
          Print #1, "{design_reference.deck_and_bulkhead_layout.bulkheads} new geom_placeholder
regular_frame_spacing_fwd_of_MMR_spaces"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s} new placeholder overall"
            Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.overall} new var fwd_end"
              Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.overall.fwd_end} set_units_category (length_default)"
```

```
            Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.overall} new var frame_spacing"
              Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.overall.frame_spacing} set_units_category (length_default)"
                Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.overall.frame_spacing} = user_variables.frame_spacing"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s} new placeholder compartment_length"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s} new placeholder bulkhead_position"
        Print #1, "{design_reference.deck_and_bulkhead_layout.bulkheads} new geom_placeholder
regular_frame_spacing_MMR_spaces"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces} new
placeholder overall"
            Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.overall
} new var fwd_end"
              Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.overall.
fwd_end} set_units_category (length_default)"
            Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.overall
} new var frame_spacing"
              Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.overall.
frame_spacing} set_units_category (length_default)"
                Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.overall.
frame_spacing} = user_variables.frame_spacing"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces} new
placeholder compartment_length"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces} new
placeholder bulkhead_position"
        Print #1, "{design_reference.deck_and_bulkhead_layout.bulkheads} new geom_placeholder
regular_frame_spacing_aft_of_MMR_spaces"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces}
new placeholder overall"
```

```
            Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
overall} new var fwd_end"
              Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
overall.fwd_end} set_units_category (length_default)"
            Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
overall} new var frame_spacing"
              Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
overall.frame_spacing} set_units_category (length_default)"
                Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
overall.frame_spacing} = user_variables.frame_spacing"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces}
new placeholder compartment_length"
          Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces}
new placeholder bulkhead_position"
      Print #1, "{design_reference.deck_and_bulkhead_layout} new layout_grid layout_grid"
          Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.length} =
user_variables.LOA"
          Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.beam} =
user_variables.Beam"
          Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.depth} =
user_variables.HOA"
      Print #1, "{design_reference.deck_and_bulkhead_layout} new geom_placeholder planes"
          Print #1, "{design_reference.deck_and_bulkhead_layout.planes} new geom_placeholder Z"
      Print #1, "{design_reference.deck_and_bulkhead_layout} new geom_placeholder decks"

      'origin points & key nodes
      Print #1, "{design_reference.origin_points_and_key_nodes} new point origin"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point AP"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point AP_wl"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point midships"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point FP"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point FP_wl"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point min_x"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point max_x"
      Print #1, "{design_reference.origin_points_and_key_nodes} new point max_x_max_z"

      Print #1, "{design_reference.profiles_sheets_clearances_etc} new geom_placeholder
sightline_over_bow"
         Print #1, "{design_reference.profiles_sheets_clearances_etc.sightline_over_bow} new var
view_distance_in_Lwl"
```

```
        Print #1, "{design_reference.profiles_sheets_clearances_etc.sightline_over_bow} new var
waterline_length"
            Print #1,
"{design_reference.profiles_sheets_clearances_etc.sightline_over_bow.waterline_length}
set_units_category (length_default)"
        Print #1, "{design_reference.profiles_sheets_clearances_etc.sightline_over_bow} new polyline
sightline"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.sightline_over_bow.sightline} new
point p1"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.sightline_over_bow.sightline} new
point p2"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.sightline_over_bow.sightline} new
point p3"
      Print #1, "{design_reference.profiles_sheets_clearances_etc} new geom_placeholder profiles"
        Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles} new geom_placeholder
points"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
profile_point_one"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
profile_point_two"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
transverse_point_one"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
transverse_point_two"
        Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles} new geom_placeholder
sheets"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets} new sheet
wind_profile_sheet"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets} new sheet
upper_deck_sheet"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets} new sheet
margin_sheet"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets} new sheet
transverse_sheet"

      Print #1, "{design_reference.envelope} new geom_placeholder hull_generation"
        Print #1, "{design_reference.envelope.hull_generation} new placeholder quickhull_model"
            Print #1, "{design_reference.envelope.hull_generation.quickhull_model} new
geom_placeholder hull_library"
                Print #1, "{design_reference.envelope.hull_generation.quickhull_model.hull_library} new
quickhull0 qh0"
            Print #1, "{design_reference.envelope.hull_generation.quickhull_model} new
geom_placeholder Actual_scaling_model"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model} new var
reference_LBP"
```

```
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.reference_LBP}
set_units_category (length_default)"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model} new var
reference_beam"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.reference_beam}
set_units_category (length_default)"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model} new var
reference_depth"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.reference_depth}
set_units_category (length_default)"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model} new
geom_placeholder controled_hull"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new CSA_param CSA_param"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new quickhull1 qh1"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new hull_surfaces2 hull_surface"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new solid_body hull"
                    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hul
l} from_bounds
(design.reference.envelopehull_generation.quichhull_model.Actual_scaling_model.controled_hull.hull_
surface.bounds"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new sheet deck"
                    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.dec
k} sheet_from_surface
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.upper_deck)"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new sheet stbd_hull"
                    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.stb
d_hull} sheet_from_surface
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.stbd_hull)"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new sheet port_hull"
              Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.por
t_hull} sheet_from_surface
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.port_hull)"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new sheet transom"
              Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.tra
nsom} sheet_from_surface
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.transom)"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new sheet bow"
              Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.bo
w} sheet_from_surface
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.bow)"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull}
new sheet keel"
              Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.kee
l} sheet_from_surface
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.keel)"
          Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model} new
geom_placeholder checks"
            'will leave this folder empty for the moment with the intent of developing my own checks
as I determine what is needed
          Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model} new
geom_placeholder underwater_form"
            Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.underwater_form}
new solid_body hull_divided"
```

```
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.underwater_form}
new plane water_plane"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.underwater_form}
new solid_body underwater_hull"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.underwater_form}
new point centroid"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.underwater_form}
new sheet midships"
            Print #1, "{design_reference.envelope.hull_generation.quickhull_model} new
geom_placeholder Output"
              Print #1, "{design_reference.envelope.hull_generation.quickhull_model.Output} new
geom_placeholder live_visualization"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Output.live_visualization} new
geometry_chopper Regular_bulkheads"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Output.live_visualization} new
geometry_chopper Regular_decks"
              Print #1, "{design_reference.envelope.hull_generation.quickhull_model.Output} new
geom_placeholder lines_plan_drawing"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Output.lines_plan_drawing} new
drawing aft_sections"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Output.lines_plan_drawing} new
drawing fwd_sections"
                Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Output.lines_plan_drawing} new
drawing All_sections"
         Print #1, "{design_reference.envelope.hull_generation} new geom_placeholder
total_bouyant_bodies"
           Print #1, "{design_reference.envelope.hull_generation.total_bouyant_bodies} new solid_body
hull_solid_2_bouyant_body"

    Print #1, "{design_reference.envelope} new geom_placeholder superstructure_modeling"
      Print #1, "{design_reference.envelope.superstructure_modeling} new solid_body SS_envelope"

    Print #1, "{design_reference.envelope} new geom_placeholder flight_deck_cutting_block"

''''''''''''''''''''''''''''''
'Insert design tertiary files'
''''''''''''''''''''''''''''''

    Print #1, "{design.envelope} new geom_placeholder solid_model"
```

```
        Print #1, "{design.envelope.solid_model} new solid_body hull"
'       Print #1, "{design.envelope.solid_model} new solid_body superstructure"
        Print #1, "{design.envelope.solid_model} new geom_placeholder hull_sheets"
          Print #1, "{design.envelope.solid_model.hull_sheets} new sheet deck"
            Print #1, "{design.envelope.solid_model.hull_sheets.deck} copy_of
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.dec
k)"
          Print #1, "{design.envelope.solid_model.hull_sheets} new sheet stbd_hull"
            Print #1, "{design.envelope.solid_model.hull_sheets.stbd_hull} copy_of
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.stbd
_hull)"
          Print #1, "{design.envelope.solid_model.hull_sheets} new sheet port_hull"
            Print #1, "{design.envelope.solid_model.hull_sheets.port_hull} copy_of
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.port
_hull)"
          Print #1, "{design.envelope.solid_model.hull_sheets} new sheet transom"
            Print #1, "{design.envelope.solid_model.hull_sheets.transom} copy_of
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.tran
som)"
          Print #1, "{design.envelope.solid_model.hull_sheets} new sheet bow"
            Print #1, "{design.envelope.solid_model.hull_sheets.bow} copy_of
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.bow
)"
          Print #1, "{design.envelope.solid_model.hull_sheets} new sheet keel"
            Print #1, "{design.envelope.solid_model.hull_sheets.keel} copy_of
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.keel
)"
        Print #1, "{design.envelope.solid_model} new geom_placeholder hull_bulkhead_sheets"
        Print #1, "{design.envelope.solid_model} new geom_placeholder hull_deck_sheets"
        Print #1, "{design.envelope.solid_model} new geom_placeholder deckhouse_deck_sheets"
        Print #1, "{design.envelope.solid_model} new geom_placeholder deckhouse_bulkhead_sheets"

      Print #1, "{design.design_building_block_model} new building_block Design"
        Print #1, "{design.design_building_block_model.Design.attributes.use_sub_blocks_ignore_this}"
        Print #1,
"{design.design_building_block_model.Design.requirements.attributes.use_sub_blocks_ignore_this}"
        Print #1, "{design.design_building_block_model.Design} new building_block requirements"
          Print #1,
"{design.design_building_block_model.Design.requirements.attributes.use_sub_blocks_ignore_this}"
          Print #1, "{design.design_building_block_model.Design.requirements} new building_block
speed"
          Print #1, "{design.design_building_block_model.Design.requirements} new building_block
range"
          Print #1, "{design.design_building_block_model.Design.requirements} new building_block
endurance"
          Print #1, "{design.design_building_block_model.Design.requirements} new building_block
complement"
```

```
        Print #1, "{design.design_building_block_model.Design.requirements} new building_block
user_defined"
        Print #1, "{design.design_building_block_model.Design.requirements} new building_block
powering"
        Print #1, "{design.design_building_block_model.Design.requirements} new building_block
electric_load"
        Print #1, "{design.design_building_block_model.Design.requirements} new building_block fuel"
          Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics} new var
propulsion_power_at_endurance_speed"
            Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.propulsion_p
ower_at_endurance_speed} set_units_category (power)"
          Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics} new var
electrical_power_at_endurance_speed"
            Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.electrical_po
wer_at_endurance_speed} set_units_category (power)"
          Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics} new var
total_power_at_endurance_speed"
            Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.total_power_
at_endurance_speed} set_units_category (power)"
          Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics} new var
time_at_endurance_speed"
            Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.time_at_end
urance_speed} set_units_category (time_major)"
          Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics} new var
SFC_at_endurance_speed"
            Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.SFC_at_endu
rance_speed} set_units_category (mass_per_energy)"
          Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics} new var
margins"
            Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.margins}
set_units_category (proportion)"
          Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics} new
char_consumable fuel_demand_for_endurance_speed"
        Print #1, "{design.design_building_block_model.Design.requirements} new building_block
structure"
```

```
            Print #1,
"{design.design_building_block_model.Design.requirements.structure.attributes.use_sub_blocks_ignore
_this}"
            Print #1, "{design.design_building_block_model.Design.requirements.structure} new
building_block overall_estimate"



            Set Method = Range("C68")
            Set Material = Range("C69")
            Set z_cent_est = Range("C35")

        '       Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.use_s
ub_blocks_ignore_this}"
              Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var method"
                Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.method} = " & Method
              Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var material"
                Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.material} = " & Material
              Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var weight_margin"
                Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.weight_margin} =
(1+(setup.margins.design_margin_multipliers.weight_margin))*(1+(setup.margins.design_margin_multi
pliers.weight_allowance))"
              Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var x_position"
                Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.x_position} set_units_category (length_default)"
                Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.x_position} = (design.envelope.solid_model.hull.attributes.centroid.x)"
              Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var z_position"
```

```
                    Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.z_position} set_units_category (length_default)"
                    Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.z_position} = (user_variables.Depth_midships)*" & z_cent_est
                  Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var length"
                    Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.length} set_units_category (length_default)"
                  Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var beam"
                    Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.beam} set_units_category (length_default)"
                  Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new var depth"
                    Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.depth} set_units_category (length_default)"
            Print #1, "{design.design_building_block_model.Design} new building_block deckhouse"
              Print #1,
"{design.design_building_block_model.Design.deckhouse.attributes.use_sub_blocks_ignore_this}"
            Print #1, "{design.design_building_block_model.Design} new building_block MMRs"
              Print #1,
"{design.design_building_block_model.Design.MMRs.attributes.use_sub_blocks_ignore_this}"
              Print #1, "{design.design_building_block_model.Design.MMRs} new building_block
exhaust_stacks"
                Print #1,
"{design.design_building_block_model.Design.MMRs.exhaust_stacks.attributes.use_sub_blocks_ignore_
this}"

            Print #1, "{design.design_building_block_model.Design} new building_block equipment"
              Print #1,
"{design.design_building_block_model.Design.equipment.attributes.use_sub_blocks_ignore_this}"
            'might need more subblocks under design.  Will work out as I develop model

          Print #1, "{design.systems} new concept_placeholder service_highways"

          Print #1, "{design.appendages} new apendage_container appendages"


''''''''''''''''''''''''''''
'Insert Audit tertiary files'
```

```
''''''''''''''''''''''''''

'       Print #1, "{audit.weight} new concept_placeholder weight_groups"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_1"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_2"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_3"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_4"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_5"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_6"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_7"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_8"
'          Print #1, "{audit.weight.weight_groups} new weight_summary GP_unallocated"

        Print #1, "{audit.weight} new placeholder simplifications"
           Print #1, "{audit.weight.simplifications} new var fuel"
              Print #1, "{audit.weight.simplifications.fuel} set_units_category (mass_default)"
           Print #1, "{audit.weight.simplifications} new var fuel_margin"
              Print #1, "{audit.weight.simplifications.fuel_margin} set_units_category (mass_default)"

        Print #1, "{audit.weight} new placeholder margins"
           Print #1, "{audit.weight.margins} new var lightship"
            ' Print #1, "{audit.weight.margins.lightship} set_units_category (mass_default)"
           Print #1, "{audit.weight.margins} new var design_margin"
              Print #1, "{audit.weight.margins.design_margin} set_units_category (mass_default)"
           Print #1, "{audit.weight.margins} new var growth_allowance"
              Print #1, "{audit.weight.margins.growth_allowance} set_units_category (mass_default)"

        Print #1, "{audit.weight} new concept_placeholder lightship"
           Print #1, "{audit.weight.lightship} new block_summary block_summary"
              Print #1, "{audit.weight.lightship.block_summary.top_level_block} ->
design.design_building_block_model.Design"
           Print #1, "{audit.weight.lightship} new block_definition block_definition_light"
              Print #1, "{audit.weight.lightship.block_definition_light.block_summary} ->
audit.weight.lightship.block_summary"
           Print #1, "{audit.weight.lightship} new design_audit design_audit"
              Print #1, "{audit.weight.lightship.design_audit.block_definition} ->
audit.weight.lightship.block_definition_light"
              Print #1, "{audit.weight.lightship.design_audit.characteristic.weight}"
              Print #1, "{audit.weight.lightship.design_audit.characteristic.condition} ->
setup.ship_conditions.lightship_condition"
              Print #1, "{audit.weight.lightship.design_audit.hierarchy.classification}"
              Print #1, "{audit.weight.lightship.design_audit.hierarchy_level.all}"

        Print #1, "{audit.weight} new concept_placeholder lightship_including_margins"
           Print #1, "{audit.weight.lightship_including_margins} new block_summary block_summary"
              Print #1, "{audit.weight.lightship_including_margins.block_summary.top_level_block} ->
design.design_building_block_model.Design"
```

```
        Print #1, "{audit.weight.lightship_including_margins} new block_definition
block_definition_light_with_margins"
          Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.block_summary} ->
audit.weight.lightship_including_margins.block_summary"
        Print #1, "{audit.weight.lightship_including_margins} new design_audit design_audit"
          Print #1, "{audit.weight.lightship_including_margins.design_audit.block_definition} ->
audit.weight.lightship_including_margins.block_definition_light_with_margins"
          Print #1, "{audit.weight.lightship_including_margins.design_audit.characteristic.weight}"
          Print #1, "{audit.weight.lightship_including_margins.design_audit.characteristic.condition} ->
setup.ship_conditions.lightship_condition"
          Print #1, "{audit.weight.lightship_including_margins.design_audit.hierarchy.classification}"
          Print #1, "{audit.weight.lightship_including_margins.design_audit.hierarchy_level.all}"

      Print #1, "{audit.weight} new concept_placeholder simplified_full_load"
        Print #1, "{audit.weight.simplified_full_load} new block_summary block_summary"
          Print #1, "{audit.weight.simplified_full_load.block_summary.top_level_block} ->
design.design_building_block_model.Design"
        Print #1, "{audit.weight.simplified_full_load} new block_definition block_definition_full_load"
          Print #1, "{audit.weight.simplified_full_load.block_definition_full_load.block_summary} ->
audit.weight.simplified_full_load.block_summary"
        Print #1, "{audit.weight.simplified_full_load} new design_audit design_audit"
          Print #1, "{audit.weight.simplified_full_load.design_audit.block_definition} ->
audit.weight.simplified_full_load.block_definition_full_load"
          Print #1, "{audit.weight.simplified_full_load.design_audit.characteristic.weight}"
          Print #1, "{audit.weight.simplified_full_load.design_audit.characteristic.condition} ->
setup.ship_conditions.simplified_full_load_condition"
          Print #1, "{audit.weight.simplified_full_load.design_audit.hierarchy.classification}"
          Print #1, "{audit.weight.simplified_full_load.design_audit.hierarchy_level.all}"

      Print #1, "{audit.weight} new concept_placeholder full_load"
        Print #1, "{audit.weight.full_load} new block_summary block_summary"
          Print #1, "{audit.weight.full_load.block_summary.top_level_block} ->
design.design_building_block_model.Design"
        Print #1, "{audit.weight.full_load} new block_definition block_definition_full_load"
          Print #1, "{audit.weight.full_load.block_definition_full_load.block_summary} ->
audit.weight.full_load.block_summary"
        Print #1, "{audit.weight.full_load} new design_audit design_audit"
          Print #1, "{audit.weight.full_load.design_audit.block_definition} ->
audit.weight.full_load.block_definition_full_load"
          Print #1, "{audit.weight.full_load.design_audit.characteristic.weight}"
          Print #1, "{audit.weight.full_load.design_audit.characteristic.condition} ->
setup.ship_conditions.full_load_condition"
          Print #1, "{audit.weight.full_load.design_audit.hierarchy.classification}"
          Print #1, "{audit.weight.full_load.design_audit.hierarchy_level.all}"

      Print #1, "{audit.weight} new concept_placeholder full_load_including_margins"
        Print #1, "{audit.weight.full_load_including_margins} new block_summary block_summary"
```

```
            Print #1, "{audit.weight.full_load_including_margins.block_summary.top_level_block} ->
design.design_building_block_model.Design"
          Print #1, "{audit.weight.full_load_including_margins} new block_definition
block_definition_full_load_with_margins"
            Print #1,
"{audit.weight.full_load_including_margins.block_definition_full_load_with_margins.block_summary} ->
audit.weight.full_load_including_margins.block_summary"
          Print #1, "{audit.weight.full_load_including_margins} new design_audit design_audit"
            Print #1, "{audit.weight.full_load_including_margins.design_audit.block_definition} ->
audit.weight.full_load_including_margins.block_definition_full_load_with_margins"
            Print #1, "{audit.weight.full_load_including_margins.design_audit.characteristic.weight}"
            Print #1, "{audit.weight.full_load_including_margins.design_audit.characteristic.condition} ->
setup.ship_conditions.full_load_condition"
            Print #1, "{audit.weight.full_load_including_margins.design_audit.hierarchy.classification}"
            Print #1, "{audit.weight.full_load_including_margins.design_audit.hierarchy_level.all}"

        Print #1, "{audit.space} new concept_placeholder space_by_classification"
          Print #1, "{audit.space.space_by_classification} new block_summary block_summary"
          Print #1, "{audit.space.space_by_classification} new block_definition block_definition_space"
          Print #1, "{audit.space.space_by_classification} new design_audit design_audit"

        Print #1, "{audit.space} new concept_placeholder space_by_block"
          Print #1, "{audit.space.space_by_block} new block_summary block_summary"
          Print #1, "{audit.space.space_by_block} new block_definition block_definition_space"
          Print #1, "{audit.space.space_by_block} new design_audit design_audit"


        Print #1, "{audit.discrete_audits} new concept_placeholder weight_by_block"
          Print #1, "{audit.discrete_audits.weight_by_block} new block_summary block_summary"
          Print #1, "{audit.discrete_audits.weight_by_block} new block_definition
block_definition_allocated_weight_full_load"
          Print #1, "{audit.discrete_audits.weight_by_block} new design_audit design_audit"

        Print #1, "{audit.discrete_audits} new concept_placeholder complement"
          Print #1, "{audit.discrete_audits.complement} new block_summary block_summary"
          Print #1, "{audit.discrete_audits.complement} new block_definition
block_definition_user_defined_variables"
          Print #1, "{audit.discrete_audits.complement} new design_audit design_audit"

        Print #1, "{audit.discrete_audits} new concept_placeholder user_defined_variables_by_block"
          Print #1, "{audit.discrete_audits.user_defined_variables_by_block} new block_summary
block_summary"
          Print #1, "{audit.discrete_audits.user_defined_variables_by_block} new block_definition
block_definition_user_defined_variables"
          Print #1, "{audit.discrete_audits.user_defined_variables_by_block} new design_audit
design_audit"

        Print #1, "{audit.discrete_audits} new concept_placeholder group_weights"
        Print #1, "{audit.discrete_audits.group_weights} new block_summary block_summary"
        Print #1, "{audit.discrete_audits.group_weights} new block_definition 
block_definition_allocated_weight_full_load"
        Print #1, "{audit.discrete_audits.group_weights} new design_audit design_audit"

      Print #1, "{audit.discrete_audits} new concept_placeholder consumables_by_block"
        Print #1, "{audit.discrete_audits.consumables_by_block} new block_summary block_summary"
        Print #1, "{audit.discrete_audits.consumables_by_block} new block_definition block_definition"
        Print #1, "{audit.discrete_audits.consumables_by_block} new design_audit design_audit"

      Print #1, "{audit.discrete_audits} new concept_placeholder services"
        Print #1, "{audit.discrete_audits.services} new block_summary block_summary"
        Print #1, "{audit.discrete_audits.services} new block_definition block_definition"
        Print #1, "{audit.discrete_audits.services} new design_audit design_audit"


      Print #1, "{audit.clash_detection} new clash_detection clash_detection"


      Print #1, "{audit.interblock_relationships} new node_relationships node_relationships"

      Print #1, "{audit.infringements} new design_infringements infringements"

      Print #1, "{audit.iteation_switch} new placeholder weight_vs_displacement_switch"
        Print #1, "{audit.iteration_switch.weight_vs_displacement_switch} new var 
input_estimated_full_load_displacement_without_margins"
          Print #1, 
"{audit.iteration_switch.weight_vs_displacement_switch.input_estimated_full_load_displacement_with
out_margins} set_units_category (mass_default)"
        Print #1, "{audit.iteration_switch.weight_vs_displacement_switch} new var 
input_unallocated_weight_full_load"
          Print #1, 
"{audit.iteration_switch.weight_vs_displacement_switch.input_unallocated_weight_full_load} 
set_units_category (mass_default)"
        Print #1, "{audit.iteration_switch.weight_vs_displacement_switch} new var 
output_full_load_displacement_without_margins"
          Print #1, 
"{audit.iteration_switch.weight_vs_displacement_switch.output_full_load_displacement_without_marg
ins} set_units_category (mass_default)"
        Print #1, "{audit.iteration_switch.weight_vs_displacement_switch} new var 
output_unallocated_weight_full_load"
          Print #1, 
"{audit.iteration_switch.weight_vs_displacement_switch.output_unallocated_weight_full_load} 
set_units_category (mass_default)"

      Print #1, "{audit.iteation_switch} new placeholder space_comparison"
        Print #1, "{audit.iteation_switch.space_comparison} new placeholder outputs"
```

```
            Print #1, "{audit.iteation_switch.space_comparison.outputs} new var
assumed_deck_head_height"
              Print #1, "{audit.iteation_switch.space_comparison.outputs.assumed_deck_head_height}
set_units_category (length_default)"
            Print #1, "{audit.iteation_switch.space_comparison.outputs} new var
total_available_volume_from_solid_model"
              Print #1,
"{audit.iteation_switch.space_comparison.outputs.total_available_volume_from_solid_model}
set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.outputs} new var
total_internal_volume_supply_from_building_block_model"
              Print #1,
"{audit.iteation_switch.space_comparison.outputs.total_internal_volume_supply_from_building_block_
model} set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.outputs} new var
total_internal_volume_demand_from_building_block_model"
              Print #1,
"{audit.iteation_switch.space_comparison.outputs.total_internal_volume_demand_from_building_bloc
k_model} set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.outputs} new var
machinery_space_volume"
              Print #1, "{audit.iteation_switch.space_comparison.outputs.machinery_space_volume}
set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.outputs} new var
consumable_tanks_volume"
              Print #1, "{audit.iteation_switch.space_comparison.outputs.consumable_tanks_volume}
set_units_category (volume_default)"
          Print #1, "{audit.iteation_switch.space_comparison} new placeholder inputs"
            Print #1, "{audit.iteation_switch.space_comparison.inputs} new var design_gross_volume"
              Print #1, "{audit.iteation_switch.space_comparison.inputs.design_gross_volume}
set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.inputs} new var
design_machinery_volume"
              Print #1, "{audit.iteation_switch.space_comparison.inputs.design_machinery_volume}
set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.inputs} new var design_tanks_volume"
              Print #1, "{audit.iteation_switch.space_comparison.inputs.design_tanks_volume}
set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.inputs} new var design_void_volume"
              Print #1, "{audit.iteation_switch.space_comparison.inputs.design_void_volume}
set_units_category (volume_default)"
            Print #1, "{audit.iteation_switch.space_comparison.inputs} new var design_nett_volume"
              Print #1, "{audit.iteation_switch.space_comparison.inputs.design_nett_volume}
set_units_category (volume_default)"

''''''''''''
'Moving Files'
''''''''''''
```

```
Print #1, "{Concept.consumables.fresh_water} cut"
Print #1, "{setup.consumable_densities} paste"
Print #1, "{Concept.consumables.lube_oil} cut"
Print #1, "{setup.consumable_densities} paste"
Print #1, "{Concept.ANALYSIS.stability} cut"
Print #1, "{analysis} paste"
Print #1, "{Concept.audit} cut"
Print #1, "{audit} paste"
Print #1, "{Concept} delete"

''''''''''''''''''
'Setting up pointers'
''''''''''''''''''

  Print #1, "{design.envelope.solid_model.hull} from_bounds
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.bounds)"

Worksheets("Import_to_Paramarine").Activate
  Call ExportKCL(output_file)
End Sub
```

Code for Initial Hull and Equipment Insertion

```
Sub create_ship()



Application.Calculation = xlAutomatic

Application.Calculation = xlManual

 'ordering_bulkheads

 Sheets("References").Select
 Range("AD4:AE34").Select
 Selection.Copy
 ActiveWindow.SmallScroll Down:=-12
 Range("U4").Select
 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
   :=False, Transpose:=False
 Range("AF4:AG43").Select
 Application.CutCopyMode = False
 Selection.Copy
 ActiveWindow.SmallScroll Down:=-18
 Range("X4").Select
 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
   :=False, Transpose:=False
 Range("AH4:AI34").Select
 Application.CutCopyMode = False
 Selection.Copy
 ActiveWindow.SmallScroll Down:=-12
 Range("AA4").Select
 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
   :=False, Transpose:=False
 Range("V4:V34").Select
 Application.CutCopyMode = False
 ActiveWorkbook.Worksheets("References").Sort.SortFields.Clear
 ActiveWorkbook.Worksheets("References").Sort.SortFields.Add Key:=Range("V4") _
   , SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal
 With ActiveWorkbook.Worksheets("References").Sort
   .SetRange Range("U3:V34")
   .Header = xlYes
   .MatchCase = False
   .Orientation = xlTopToBottom
   .SortMethod = xlPinYin
   .Apply
 End With
 ActiveWindow.SmallScroll Down:=-9
 Range("Y4:Y43").Select
 ActiveWorkbook.Worksheets("References").Sort.SortFields.Clear
```

```
   ActiveWorkbook.Worksheets("References").Sort.SortFields.Add Key:=Range("Y4") _
      , SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets("References").Sort
      .SetRange Range("X3:Y43")
      .Header = xlYes
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
   End With
   Range("AB4:AB34").Select
   ActiveWorkbook.Worksheets("References").Sort.SortFields.Clear
   ActiveWorkbook.Worksheets("References").Sort.SortFields.Add Key:=Range("AB4") _
      , SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets("References").Sort
      .SetRange Range("AA3:AB34")
      .Header = xlYes
      .MatchCase = False
      .Orientation = xlTopToBottom
      .SortMethod = xlPinYin
      .Apply
   End With

Application.Calculation = xlAutomatic

   Call find_files
      'find_files is in Generic Functions
      Call open_output_file(output_file)
      Print #1, "deselect all"

      Print #1, "{fixed_data.units.length_default.ft}"
      Print #1, "{fixed_data.units.mass_default.LT}"
      Print #1, "{fixed_data.units.volume_default.ft3}"
      Print #1, "{fixed_data.units.density.ft3/LT}"

''''''''''''''''''''''''
'Initial Input Parameters'
''''''''''''''''''''''''

Worksheets("General_Info").Activate

sea_water_den = Range("E24")
fuel_density = Range("E29")

        Print #1, "{setup.consumable_densities.sea_water} = " & sea_water_den; "[kg/m3]"
        Print #1, "{setup.consumable_densities.dieso_fuel} = " & fuel_density; "[lb/ft3]"

 Worksheets("Output").Activate
```

```
    Set stern_loc = Range("G12")
    Set length = Range("G3")
    Set Beam = Range("G9")
    Set Depth = Range("G5")
    Set LOA = Range("G10")
    Set HOA = Range("G11")
    Set Draft = Range("G14")
    Set Ships_Range = Range("G17")
    Set Endurance_Speed = Range("G18")
    Set Max_Speed = Range("G19")
    Set C_p = Range("G20")
    Set C_m = Range("G21")
    Set Hull_Addendum_Height = Range("C14")
    Set min_depth = Range("G7")

    Print #1, "{user_variables.LBP} = ", length, "[ft]"
    Print #1, "{user_variables.Beam} = ", Beam, "[ft]"
    Print #1, "{user_variables.Depth_midships} = ", Depth, "[ft]"
    Print #1, "{user_variables.Depth_scaled_from_parent_hull} = ", min_depth, "[ft]"
    Print #1, "{user_variables.LOA} = ", LOA, "[ft]"
    Print #1, "{user_variables.HOA} = ", HOA, "[ft]"
    Print #1, "{user_variables.Draft} = ", Draft, "[ft]"
    Print #1, "{user_variables.Range} = ", Ships_Range, "[nm]"
    Print #1, "{user_variables.Endurance_Speed} = ", Endurance_Speed, "[kt]"
    Print #1, "{user_variables.Max_Speed} = ", Max_Speed, "[kt]"
    Print #1, "{user_variables.prismatic_coefficient} = ", C_p
    Print #1, "{user_variables.midship_coefficient} = ", C_m

''''''''''''
'Insert Hull'
''''''''''''

    'key point insertion

    Worksheets("Chosen_Hull").Activate

    Set Scaling_Factor = Range("G4:G30")
    Set key_points = Range("F4:F30")
    Set key_points_coord = Range("C4:C30")
    Set key_points_location = Range("B4:B30")

    For i = 1 To 27
      Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.key_points.";
key_points_location(i); "."; key_points_coord(i); "} = "; key_points(i); "[ft]"
    Next i
    'guide curve insertion

    Set Scaling_Factor_gc = Range("P4:P227")
    Set guide_curves = Range("O4:O227")
    Set guide_curves_coord = Range("L4:L227")
    Set guide_curves_location = Range("J4:J227")
    Set guide_curves_ref = Range("K4:K227")

    For i = 1 To 224
      Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.guide_curves.";
guide_curves_location(i); ".control_points."; guide_curves_ref(i); "."; guide_curves_coord(i); "} = ";
guide_curves(i); "[ft]"
    Next i

    'csa_param entries

    Set ref_length = Range("D39")
    Set ref_beam = Range("D40")
    Set ref_depth = Range("D41")

    Set entry_coeff = Range("D34")
    Set run_coeff = Range("D35")
    Set skeg_area_coeff = Range("D36")
    Set transom_area_coeff = Range("D37")
    Set bow_area_coeff = Range("D38")

    Set ref_dispt_volume = Range("D42")
    dispt_volume = ref_dispt_volume
    Set ref_LCB_from_midship = Range("D43")
    LCB_from_midship = ref_LCB_from_midship
    midship = ref_length / 2
    LCB = midship + LCB_from_midship
    waterline_aft = 0
    waterline_fwd = ref_length
    Set ref_waterplane_z = Range("D46")
    Set ref_waterplane_d = Range("D47")
    waterplane_z = ref_waterplane_z
    waterplane_d = ref_waterplane_d

    Worksheets("Defaults").Activate
  '  Set C_p = Range("C11")
  '  Set C_m = Range("C12")

    'inserting reference values
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.reference_LBP} =
"; ref_length; "[ft]"
```

```
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.reference_beam}
= "; ref_beam; "[ft]"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.reference_depth}
= "; ref_depth; "[ft]"

    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.Cp} = (user_variables.prismatic_coefficient)"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.Cm} = (user_variables.midship_coefficient)"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.dispt_volume} = "; ref_dispt_volume; "[ft3]"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.waterline_aft_x} = "; waterline_aft; "[ft]"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.pmb_aft_x} =
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.key_points.pmb_aft_btm.
x"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.pmb_fwd_x} =
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.key_points.pmb_fwd_btm
.x"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.waterline_fwd_x} = "; waterline_fwd; "[ft]"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.entry_coeff} = "; entry_coeff
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.run_coeff} = "; run_coeff
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.skeg_area_coeff} = "; sked_area_coeff
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.transom_area_coeff} = "; transom_area_coeff
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.bow_area_coeff} = "; bow_area_coeff
```

```
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.LCB} = "; LCB; "[ft]"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CS
A_param.input.aft_cut_up_x} =
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.key_points.aft_cut_up.x"

    'Populating Q1

    For i = 1 To 27
      Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.key_points."; key_points_location(i); "."; key_points_coord(i); "} =
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.key_points.";
key_points_location(i); "."; key_points_coord(i)
    Next i

    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.guide_curves.transom} ->
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.guide_curves.midships} ->
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.guide_curves.bow} -> design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.guide_curves.keel_aft} ->
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.guide_curves.deck_aft} ->
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.guide_curves.keel_fwd} ->
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.guide_curves.deck_fwd} ->
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.target_CSA} ->
```

```
design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.CSA_
param"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.num_CSA_iterations} = 20"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.waterplane.z} = "; waterplane_z
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh
1.waterplane.d} = "; waterplane_d; "[ft]"

    'Creating Hull

    Print #1, "{user_variables.daughter_parent_hull_scaling} = " & length / ref_length

    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hul
l_surface.port_hull_surface} ->
design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.qh1.
output_surface"
    Print #1,
"{design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hul
l} from_bounds
(design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.controled_hull.hull
_surface.bounds )"

    Print #1, "{design.envelope.solid_model.hull} scale
(user_variables.LBP/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.
reference_LBP,
user_variables.Beam/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_mode
l.reference_beam,
user_variables.Depth_scaled_from_parent_hull/design_reference.envelope.hull_generation.quickhull_
model.Actual_scaling_model.reference_depth, references.origin.origin)"
    Print #1, "{design.envelope.solid_model.hull_sheets.deck} scale
(user_variables.LBP/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.
reference_LBP,
user_variables.Beam/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_mode
l.reference_beam,
user_variables.Depth_scaled_from_parent_hull/design_reference.envelope.hull_generation.quickhull_
model.Actual_scaling_model.reference_depth, references.origin.origin)"
    Print #1, "{design.envelope.solid_model.hull_sheets.stbd_hull} scale
(user_variables.LBP/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.
reference_LBP,
user_variables.Beam/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_mode
l.reference_beam,
user_variables.Depth_scaled_from_parent_hull/design_reference.envelope.hull_generation.quickhull_
model.Actual_scaling_model.reference_depth, references.origin.origin)"
```

```
      Print #1, "{design.envelope.solid_model.hull_sheets.port_hull} scale
(user_variables.LBP/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.
reference_LBP,
user_variables.Beam/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_mode
l.reference_beam,
user_variables.Depth_scaled_from_parent_hull/design_reference.envelope.hull_generation.quickhull_
model.Actual_scaling_model.reference_depth, references.origin.origin)"
      Print #1, "{design.envelope.solid_model.hull_sheets.transom} scale
(user_variables.LBP/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.
reference_LBP,
user_variables.Beam/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_mode
l.reference_beam,
user_variables.Depth_scaled_from_parent_hull/design_reference.envelope.hull_generation.quickhull_
model.Actual_scaling_model.reference_depth, references.origin.origin)"
      Print #1, "{design.envelope.solid_model.hull_sheets.bow} scale
(user_variables.LBP/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.
reference_LBP,
user_variables.Beam/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_mode
l.reference_beam,
user_variables.Depth_scaled_from_parent_hull/design_reference.envelope.hull_generation.quickhull_
model.Actual_scaling_model.reference_depth, references.origin.origin)"
      Print #1, "{design.envelope.solid_model.hull_sheets.keel} scale
(user_variables.LBP/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_model.
reference_LBP,
user_variables.Beam/design_reference.envelope.hull_generation.quickhull_model.Actual_scaling_mode
l.reference_beam,
user_variables.Depth_scaled_from_parent_hull/design_reference.envelope.hull_generation.quickhull_
model.Actual_scaling_model.reference_depth, references.origin.origin)"

''''''''''''''''''''''''
'Insert Decks & Bulkheads'
''''''''''''''''''''''''


  'Decks
  Worksheets("References").Activate
      num_hull_decks = 20 'Range("M5")
      Set hull_deck_height = Range("J4:J23")
      Set hull_deck_name = Range("K4:K23")
      Set weather_deck_height = Range("J5")

  For n = 1 To num_hull_decks
      Print #1, "{user_variables.deck_heights} new var " & hull_deck_name(n)
      Print #1, "{user_variables.deck_heights."; hull_deck_name(n); "} set_units_category
(length_default)"
      Print #1, "{user_variables.deck_heights."; hull_deck_name(n); "} = "; hull_deck_height(n); "[ft]"

      Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations} new var " &
hull_deck_name(n)
```

```
        Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations." &
hull_deck_name(n); "} set_units_category (length_default)"
        Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations." &
hull_deck_name(n); "} = user_variables.deck_heights." & hull_deck_name(n)
        Print #1, "{design_reference.deck_and_bulkhead_layout.planes.Z} new plane "; hull_deck_name(n)
        Print #1, "{design_reference.deck_and_bulkhead_layout.planes.Z."; hull_deck_name(n); "}
set_units_category (length_default)"
        Print #1, "{design_reference.deck_and_bulkhead_layout.planes.Z."; hull_deck_name(n); ".d} =
design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations." & hull_deck_name(n)
        Print #1, "{design_reference.deck_and_bulkhead_layout.planes.Z."; hull_deck_name(n); ".z} = 1"

        Print #1, "{design_reference.deck_and_bulkhead_layout.deck_height} new var " &
hull_deck_name(n)
        Print #1, "{design_reference.deck_and_bulkhead_layout.deck_height."; hull_deck_name(n); "}
set_units_category (length_default)"
        Print #1, "{design_reference.deck_and_bulkhead_layout.deck_height."; hull_deck_name(n); "} =
user_variables.deck_heights." & hull_deck_name(n); ""

        If hull_deck_height(n) < weather_deck_height Then

            Print #1, "{references.deck_points} new geom_placeholder " & hull_deck_name(n)
            Print #1, "{references.deck_points." & hull_deck_name(n); "} new point one"
                Print #1, "{references.deck_points." & hull_deck_name(n); ".one.x} = .25*(user_variables.LBP)"
                Print #1, "{references.deck_points." & hull_deck_name(n); ".one.y} = -
1.1*(user_variables.Beam)"
                Print #1, "{references.deck_points." & hull_deck_name(n); ".one.z} =
(user_variables.deck_heights." & hull_deck_name(n); ")"
            Print #1, "{references.deck_points." & hull_deck_name(n); "} new point two"
                Print #1, "{references.deck_points." & hull_deck_name(n); ".two.x} = -
1.25*(user_variables.LBP)"
                Print #1, "{references.deck_points." & hull_deck_name(n); ".two.y} =
1.1*(user_variables.Beam)"
                Print #1, "{references.deck_points." & hull_deck_name(n); ".two.z} =
(user_variables.deck_heights." & hull_deck_name(n); ")"
            Print #1, "{design.envelope.solid_model.hull_deck_sheets} new sheet " & hull_deck_name(n)
            Print #1, "{design.envelope.solid_model.hull_deck_sheets." & hull_deck_name(n); "} rectangle
(references.deck_points." & hull_deck_name(n); ".one, references.deck_points." & hull_deck_name(n);
".two )"
            Print #1, "{design.envelope.solid_model.hull_deck_sheets." & hull_deck_name(n); "} intersect
(design.envelope.solid_model.hull)"

        End If

    Next n

    'Bulkheads
        Set num_MMR_BHs = Range("S4")
        Set num_fwd_BHs = Range("S7")
        Set num_aft_BHs = Range("S6")
        total_num_BHs = num_MMR_BHs + num_fwd_BHs + num_aft_BHs - 2 'two is due to double
counting of BHs between sections
        Set name_BH = Range("O4:O43")
        Set BH_loc = Range("P4:P43")
        Set aft_BH_name = Range("U4:U34")
        Set aft_BH_loc = Range("V4:V34")
        Set MMR_BH_name = Range("X4:X43")
        Set MMR_BH_loc = Range("Y4:Y14")
        Set fwd_BH_name = Range("AA4:AA34")
        Set fwd_BH_loc = Range("AB4:AB34")
        Set fwd_BH_max = Range("AB35")
        Set fwd_BH_min = Range("AB36")
        Set aft_BH_max = Range("U35")
        Set aft_BH_min = Range("V36")
        Set MMR_BH_max = Range("Y44")
        Set MMR_BH_min = Range("Y45")
        aft_comp_num = num_aft_BHs
        fwd_comp_num = num_fwd_BHs
        MMR_comp_num = num_MMR_BHs - 1


    Worksheets("General_Info").Activate
      Set alphabet = Range("U4:U55")

    'all bulkheads
    For n = 1 To 40
        If name_BH(n) = 0 Then

        Else
            Print #1, "{user_variables.bulkheads} new var "; name_BH(n)
            Print #1, "{user_variables.bulkheads."; name_BH(n); "} set_units_category (length_default)"
            Print #1, "{user_variables.bulkheads."; name_BH(n); "} = "; BH_loc(n); "[ft]"
            Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.transverse_bulkheads} new
var " & name_BH(n)
            Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.transverse_bulkheads." &
name_BH(n); "} set_units_category (length_default)"
            Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.transverse_bulkheads." &
name_BH(n); "} = user_variables.bulkheads." & name_BH(n)

            Print #1, "{references.bulkhead_points} new geom_placeholder " & name_BH(n)
            Print #1, "{references.bulkhead_points." & name_BH(n); "} new point one"
                Print #1, "{references.bulkhead_points." & name_BH(n); ".one.x} = (user_variables.bulkheads."
& name_BH(n); ")"
                Print #1, "{references.bulkhead_points." & name_BH(n); ".one.y} = -1.1*(user_variables.Beam)"
                Print #1, "{references.bulkhead_points." & name_BH(n); ".one.z} = -50[ft]"
            Print #1, "{references.bulkhead_points." & name_BH(n); "} new point two"
        Print #1, "{references.bulkhead_points." & name_BH(n); ".two.x} = (user_variables.bulkheads." 
& name_BH(n); ")"
        Print #1, "{references.bulkhead_points." & name_BH(n); ".two.y} = 1.1*(user_variables.Beam)"
        Print #1, "{references.bulkhead_points." & name_BH(n); ".two.z} = 1.1*(user_variables.HOA)"
      Print #1, "{design.envelope.solid_model.hull_bulkhead_sheets} new sheet " & name_BH(n)
      Print #1, "{design.envelope.solid_model.hull_bulkhead_sheets." & name_BH(n); "} rectangle 
(references.bulkhead_points." & name_BH(n); ".one, references.bulkhead_points." & name_BH(n); 
".two )"
      Print #1, "{design.envelope.solid_model.hull_bulkhead_sheets." & name_BH(n); "} intersect 
(design.envelope.solid_model.hull)"

    End If
  Next n



      Print #1, 
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position} new var bow"
      Print #1, 
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position.bow} set_units_category (length_default)"
      Print #1, 
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position.bow} = 
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.key_points.bow_top.x"

      Print #1, 
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position} new var FP"
      Print #1, 
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position.FP} set_units_category (length_default)"
      Print #1, 
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position.FP} = 0[ft]"



  'aft & fwd bulkheads
  For n = 1 To 30
    If aft_BH_name(n) = 0 Then

    Else
      Print #1, 
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position} new var "; aft_BH_name(n)
```

```
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position."; aft_BH_name(n); "} set_units_category (length_default)"
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position."; aft_BH_name(n); "} = user_variables.bulkheads."; aft_BH_name(n)
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
compartment_length} new var"

      End If

      If fwd_BH_name(n) = 0 Then

      Else
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position} new var "; fwd_BH_name(n)
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position."; fwd_BH_name(n); "} set_units_category (length_default)"
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.bulkhead_position."; fwd_BH_name(n); "} = user_variables.bulkheads."; fwd_BH_name(n)

      End If
    Next n
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position} new var AP"
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position.AP} set_units_category (length_default)"
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position.AP} = -user_variables.LBP"

        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position} new var transom"
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position.transom} set_units_category (length_default)"
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
bulkhead_position.transom} =
design_reference.envelope.hull_generation.quickhull_model.hull_library.qh0.key_points.transom_top.x
"
```

```
  For n = 1 To aft_comp_num

     Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
compartment_length} new var compartment_aft_"; alphabet(n)
     Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_aft_of_MMR_spaces.
compartment_length.compartment_aft_"; alphabet(n); " set_units_category (length_default)"

  Next n

  For n = 1 To fwd_comp_num

     Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.compartment_length} new var compartment_fwd"; alphabet(n)
     Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_fwd_of_MMR_space
s.compartment_length.compartment_fwd_"; alphabet(n); " set_units_category (length_default)"

  Next n

  For n = 1 To MMR_comp_num

     Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.compa
rtment_length} new var compartment_MMR"; alphabet(n)
     Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.compa
rtment_length.compartment_MMR_"; alphabet(n); " set_units_category (length_default)"

  Next n

  'MMR bulkheads
  For n = 1 To 40
     If MMR_BH_name(n) = 0 Then

     Else
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.bulkhe
ad_position} new var "; MMR_BH_name(n)
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.bulkhe
ad_position."; MMR_BH_name(n); "} set_units_category (length_default)"
        Print #1,
"{design_reference.deck_and_bulkhead_layout.bulkheads.regular_frame_spacing_MMR_spaces.bulkhe
ad_position."; MMR_BH_name(n); "} = user_variables.bulkheads."; MMR_BH_name(n)
```

```
      End If
   Next n

   Print #1, "{design_reference.deck_and_bulkhead_layout.layout_grid.keel_aft.x} = "; stern_loc; "[ft]"

' Creating Hull Addendum

   If Hull_Addendum_Height > 0 Then

      Print #1, "{design_reference.envelope} new solid_body hull_addendum"
      Print #1, "{design_reference.envelope.hull_addendum} thicken_sheet
(design.envelope.solid_model.hull_sheets.deck, (user_variables.Depth_midships-
user_variables.Depth_scaled_from_parent_hull), 0[ft])"
      Print #1, "{design.envelope.solid_model.hull} unite (design_reference.envelope.hull_addendum)"

   End If

   If Hull_Addendum_Height < 0 Then

      Print #1, "{design_reference.envelope} new solid_body hull_addendum"
      Print #1, "{design_reference.envelope.hull_addendum} thicken_sheet
(design.envelope.solid_model.hull_sheets.deck, 0[ft], -(user_variables.Depth_midships-
user_variables.Depth_scaled_from_parent_hull))"
      Print #1, "{design.envelope.solid_model.hull} subtract
(design_reference.envelope.hull_addendum)"

   End If

' Redefining Hull Sheets with Hull Addendum

   Print #1, "{design.envelope.solid_model.hull_sheets.deck} copy_of
(design_reference.envelope.hull_addendum.attributes.faces.f6)"
   Print #1, "{design_referenceprofiles_sheets_clearances_etc.profiles.sheets.upper_deck_sheet}
copy_of (design_reference.envelope.hull_addendum.attributes.faces.f6)"

''''''''''''''''''''''''
'Create Equipment Library'
''''''''''''''''''''''''



''''''''''''''''''''''
'INSERT BUILDING BLOCK'
''''''''''''''''''''''

   Worksheets("General_Info").Activate

   Dim graphic_path As Range
```

```
Set graphic_path = Range("C40")

Worksheets("Compilation").Activate

'Dim bb_name As Range
'Dim bb_type As Range
Dim bb_name, bb_type, clear_length, clear_width, _
clear_height, clearance_box_type, xmin, xmax, ymin, _
ymax, zmin, zmax, weight, Wx, Wy, Wz, graphic_location, _
graphic_name, cyclic_ref, cent_x, cent_y, _
cent_z, heat_x, heat_y, heat_z, electric_x, electric_y, _
electric_z, MR_location, class, equip_loc_x, equip_loc_y, equip_loc_z As Range

Set bb_name = Range("D27:D911")
Set bb_type = Range("C27:C911")
Set clear_length = Range("J27:J911")
Set clear_width = Range("k27:K911")
Set clear_height = Range("L27:L911")
Set clearance_box_type = Range("BM27:BM911")
Set cyclic_ref = Range("A27:A911")
Set cent_x = Range("M27:M911")
Set cent_y = Range("N27:N911")
Set cent_z = Range("O27:O911")
Set xmin = Range("P27:P911")
Set xmax = Range("Q27:Q911")
Set ymin = Range("R27:R911")
Set ymax = Range("S27:S911")
Set zmin = Range("T27:T911")
Set zmax = Range("U27:U911")
Set additional_height = Range("CF27:CF911")
Set weight = Range("V27:V911")
Set Wx = Range("W27:W911")
Set Wy = Range("X27:X911")
Set Wz = Range("Y27:Y911")
Set class = Range("CD27:CD911")
Set file_name = Range("Z27:Z911")
Set heat_x = Range("BY27:BY911")
Set heat_y = Range("BZ27:BZ911")
Set heat_z = Range("CA27:CA911")
Set electric_x = Range("CI27:CI911")
Set electric_y = Range("CJ27:CJ911")
Set electric_z = Range("CK27:CK911")
Set MR_location = Range("CL27:CL911")
Set equip_loc_x = Range("BI27:BI911")
Set equip_loc_y = Range("Bj27:Bj911")
Set equip_loc_z = Range("Bk27:Bk911")
Set MMR_equip_loc_x = Range("CN27:CN911")
```

```
    Set MMR_equip_loc_y = Range("CO27:CO911")
    Set MMR_equip_loc_z = Range("CP27:CP911")
    Set electric_frequency = Range("AW27:AW911")
    Set electric_voltage = Range("AX27:AX911")
    Set electric_power = Range("BA27:BA911")
    Set cooling_power = Range("BX27:BX911")

    For n = 1 To 384

      If bb_type(n) = 0 Then

      Else

        Print #1, "{setup.Equipment_Library." & bb_type(n); "} new equipment " & bb_name(n)
        Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".construction_geometry}
new point equipment_ref_point"
        Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".construction_geometry}
new geom_placeholder clearance_box_" & cyclic_ref(n)
        Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new solid_body clearance_box"


        Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".visible_geometry." &
bb_name(n); "} translate (setup.Equipment_Library." & _
            bb_type(n); "." & bb_name(n); ".construction_geometry.equipment_ref_point.x/4,
setup.Equipment_Library." & bb_type(n); "." & _
            bb_name(n); ".construction_geometry.equipment_ref_point.y/4, setup.Equipment_Library." &
bb_type(n); "." & _
            bb_name(n); ".construction_geometry.equipment_ref_point.z/4 )"

        If weight(n) = 0 Then

        Else

          Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".characteristics} new
char_weight weight_" & cyclic_ref(n)
          Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.weight_" & _
              cyclic_ref(n); ".classification} ->setup.classification_systems." & class(n)
          Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.weight_" & _
              cyclic_ref(n); ".weight} =" & weight(n); "[lb]"
          Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.weight_" & _
              cyclic_ref(n); ".centroid.x_offset} =" & Wx(n); "[ft]+(setup.Equipment_Library." & bb_type(n);
"." & _
              bb_name(n); ".construction_geometry.equipment_ref_point.x)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.weight_" & _
                cyclic_ref(n); ".centroid.y_offset} =" & Wy(n); "[ft]+(setup.Equipment_Library." & bb_type(n);
"." & _
                bb_name(n); ".construction_geometry.equipment_ref_point.y)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.weight_" & _
                cyclic_ref(n); ".centroid.z_offset} =" & Wz(n); "[ft]+(setup.Equipment_Library." & bb_type(n);
"." & _
                bb_name(n); ".construction_geometry.equipment_ref_point.z)"

        End If

        If cooling_power(n) = "none" Then

        Else
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".characteristics} new
char_service cooling"
   '            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.cooling.x_offset} = " & heat_x(n); "[ft]+(setup.Equipment_Library." & _
    '               bb_type(n); "." & bb_name(n); ".construction_geometry.equipment_ref_point.x)"
     '          Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.cooling.y_offset} = " & heat_y(n); "[ft]+(setup.Equipment_Library." & _
      '             bb_type(n); "." & bb_name(n); ".construction_geometry.equipment_ref_point.y)"
       '        Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.cooling.z_offset} = " & heat_z(n); "[ft]+(setup.Equipment_Library." & _
        '           bb_type(n); "." & bb_name(n); ".construction_geometry.equipment_ref_point.z)"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.cooling.demand}"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.cooling.service} -> setup.service_specs.chilled_water"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.cooling.value} = "; cooling_power(n); "[kW]"

        End If

        If electric_frequency(n) = "none" Then

        Else
                Print #1, "{setup.service_specs} new service_specification electric_" & electric_frequency(n);
"Hz_" & electric_voltage(n); "V"
                Print #1, "{setup.service_specs.electric_" & electric_frequency(n); "Hz_" &
electric_voltage(n); "V.voltage} = " & electric_voltage(n); "[V]"

                If electric_frequency(n) = "DC" Then

                    Print #1, "{setup.service_specs.electric_" & electric_frequency(n); "Hz_" &
electric_voltage(n); "V.electrical_DC"
```

```
            Else

              Print #1, "{setup.service_specs.electric_" & electric_frequency(n); "Hz_" &
electric_voltage(n); "V.electrical_AC"
              Print #1, "{setup.service_specs.electric_" & electric_frequency(n); "Hz_" &
electric_voltage(n); "V.frequency} = " & electric_frequency(n); "[Hz]"

            End If

            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".characteristics} new
char_service electric_" & cyclic_ref(n)
   '        Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.electric_" & cyclic_ref(n); ".x_offset} = " & electric_x(n); _
    '           "[ft]+(setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.x)"
    '       Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.electric_" & cyclic_ref(n); ".y_offset} = " & electric_y(n); _
     '          "[ft]+(setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)"
     '      Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.electric_" & cyclic_ref(n); ".z_offset} = " & electric_z(n); _
      '         "[ft]+(setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.electric_" & cyclic_ref(n); ".demand}"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.electric_" & cyclic_ref(n); _
              ".service} -> setup.service_specs.electric_" & electric_frequency(n); "Hz_" &
electric_voltage(n); "V"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".characteristics.electric_" & cyclic_ref(n); _
              ".value} = " & electric_power(n); "[kW]"

        End If


          If clearance_box_type(n) = "cuboid" Then

            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var length"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".length} set_units_category
(length_default)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var width"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".width} set_units_category (length_default)"
```

```
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var height"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} set_units_category
(length_default)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point centroid"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point max_x_y_z"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point min_x_y_z"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".centroid.x} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.x)+" & cent_x(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".centroid.y} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)+" & cent_y(n); "[ft]"

   '     Print #1, "new placeholder " & cent_y(n); "_" & n



            If cent_z(n) = "ref" Then

              If clear_height(n) = "ref" Then

                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} = " & _
                  additional_height(n); "[ft]"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".centroid.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)-" & (additional_height(n) / 2); "[ft]"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".max_x_y_z.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".centroid.z)+" & (additional_height(n) / 2); "[ft]"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".min_x_y_z.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".centroid.z)-" & (additional_height(n) / 2); "[ft]"

              Else
```

```
                    Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} = " & _
                        clear_height(n); "[ft]"
                    Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                        ".centroid.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)-" & _
                        (additional_height(n) + (clear_height(n) / 2)); "[ft]"
                    Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                        ".max_x_y_z.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                        ".centroid.z)+" & (clear_height(n) / 2); "[ft]"
                    Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                        ".min_x_y_z.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                        ".centroid.z)-" & (clear_height(n) / 2); "[ft]"

                End If

            Else

                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} = " & _
                    clear_height(n); "[ft]"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                    ".centroid.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)+" & cent_z(n); "[ft]"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                    ".max_x_y_z.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)+" & zmax(n); "[ft]"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                    ".min_x_y_z.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)+" & zmin(n); "[ft]"

            End If

            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".width} = " & _
                clear_width(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".length} = " & _
                clear_length(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                ".max_x_y_z.x} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.x)+" & xmax(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                ".max_x_y_z.y} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)+" & ymax(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                ".min_x_y_z.x} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.x)+" & xmin(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                ".min_x_y_z.y} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)+" & ymin(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".clearance_box} " _
                ; "cuboid (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                ".min_x_y_z, setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                ".max_x_y_z)"

        End If

        If clearance_box_type(n) = "cylinder_vert" Then

            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var radius"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius} set_units_category (length_default)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius} = " & clear_length(n) / 2; "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var height"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} set_units_category
(length_default)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} = " & clear_height(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point centroid"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point center_top"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point center_bottom"
```

```
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_top.x} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.x)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_bottom.x} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.x)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_top.y} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_bottom.y} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_top.z} = " & zmax(n); "[ft]+(setup.Equipment_Library." & bb_type(n); "." &
bb_name(n); ".construction_geometry.equipment_ref_point.z)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_bottom.z} = " & zmin(n); "[ft]+(setup.Equipment_Library." & bb_type(n); "." &
bb_name(n); ".construction_geometry.equipment_ref_point.z)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".clearance_box} " _
                  ; "cylinder (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_top, setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
                  ".center_bottom, (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius))"

          End If

          If clearance_box_type(n) = "cylinder_long" Then

            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var radius"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius} set_units_category (length_default)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius} = " & clear_width(n) / 2; "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var length"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} set_units_category
(length_default)"
```

```
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} = " & clear_length(n); "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point center_forward"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point center_aft"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_forward.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_aft.z} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_forward.y} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_aft.y} = (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_forward.x} = " & xmax(n); "[ft]+(setup.Equipment_Library." & bb_type(n); "." &
bb_name(n); ".construction_geometry.equipment_ref_point.x)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_aft.x} = " & xmin(n); "[ft]+(setup.Equipment_Library." & bb_type(n); "." &
bb_name(n); ".construction_geometry.equipment_ref_point.x)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".clearance_box} " _
              ; "cylinder (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_forward, setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".center_aft, (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius))"

          End If

 'not nessecary for initial, required analysis.  Come back to time permitting

 '          If clearance_box_type(n) = "cylinder_trans" Then
 '
  '           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var length"
```

```
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".length} set_units_category
(length_default)"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var width"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".width} set_units_category (length_default)"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var height"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} set_units_category
(length_default)"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".length} = " & clear_length(n); "[ft]"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".width} = " & clear_width(n); "[ft]"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".height} = " & clear_height(n); "[ft]"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point centroid"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point max_x_y_z"
'           Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point min_x_y_z"

'       End If

        If clearance_box_type(n) = "sphere" Then

            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new var radius"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius} set_units_category (length_default)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius} = " & clear_height(n) / 2; "[ft]"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); "} new point centroid"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".centroid.x} = " & cent_x(n);
"[ft]+(setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.x)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".centroid.y} = " & cent_y(n);
"[ft]+(setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.y)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".centroid.z} = " & cent_z(n);
```

```
"[ft]+(setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.equipment_ref_point.z)"
            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".clearance_box} " _
              ; "sphere (setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); _
              ".centroid, setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".radius))"

          End If

          If file_name(n) = 0 Then

            If n < 385 Then

              If cent_x(n) = "none" Then
              Else

                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".construction_geometry.clearance_box_" & cyclic_ref(n); ".clearance_box} copy"
                Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".visible_geometry} paste"

              End If

            End If

          Else

            Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".visible_geometry}
new solid_body " & bb_name(n)
            Print #1, "read parasolid (0, 1, 2, 1, 1, 1, 0.000000, 0.000000, 0.000000, 1.000000) " &
graphic_path; "\" & bb_type(n); "\" & file_name(n); ".x_t"
            Print #1, "{" & file_name(n); "_x_t} cut"
            Print #1, "{Graphics} paste"

          End If

        Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n); ".visible_geometry." &
bb_name(n); "} copy_of (Graphics." & file_name(n); "_x_t.solid_bodies.solid_1 )"
      End If

    Next n

''''''''''''''''''''''''''''
'Insertion of Aviation Spaces'
''''''''''''''''''''''''''''
```

```
'Hangers

   Worksheets("Output").Activate

   Dim Flight_Deck_Info, Hanger_Info As Range

   Set Flight_Deck_Info = Range("X5:X13")
   Set Hanger_Info = Range("X16:X28")

   If Flight_Deck_Info(8) = "yes" Then

      Print #1, "{setup.Equipment_Library.Aviation} new equipment flight_deck"
      Print #1, "read dxf (0, 1, 2, 1, 1, 1, 0.000000, 0.000000, 0.000000, 1.000000) " & graphic_path;
"\Hangers\" & Flight_Deck_Info(7); ".dxf"
      Print #1, "{" & Flight_Deck_Info(7); "_dxf.facet_bodies.facet_1} cut"
      Print #1, "{setup.Equipment_Library.Aviation.flight_deck.visible_geometry} paste"
      Print #1, "{" & Flight_Deck_Info(7); "_dxf} delete"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block} new solid_body cutting_block"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block} new point cutting_block_min"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block} new point cutting_block_max"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block.cutting_block_min.x} = -
1.1*(user_variables.LBP)"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block.cutting_block_max.x} = (-" &
Flight_Deck_Info(1); "*(user_variables.LBP))+(" _
         & Flight_Deck_Info(5); "[ft]/2)"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block.cutting_block_min.y} = -
(user_variables.Beam/2)-10[ft]"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block.cutting_block_max.y} =
(user_variables.Beam/2)+10[ft]"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block.cutting_block_min.z} =
(design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations." _
         & Flight_Deck_Info(3); ")+" & Flight_Deck_Info(4); "[ft]+0.01[ft]"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block.cutting_block_max.z} =
(user_variables.HOA)"
      Print #1, "{design_reference.envelope.flight_deck_cutting_block.cutting_block} cuboid
(design_reference.envelope.flight_deck_cutting_block.cutting_block_min ,
design_reference.envelope.flight_deck_cutting_block.cutting_block_max)"
      Print #1, "{design.envelope.solid_model.hull} subtract
(design_reference.envelope.flight_deck_cutting_block.cutting_block )"
      Print #1, "{design.design_building_block_model.Design.equipment} new building_block Aviation"
      Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.attributes.use_sub_blocks_ignore_th
is}"
      Print #1, "{design.design_building_block_model.Design.equipment.Aviation} new
equipment_instance Flight_Deck"
      Print #1, "{design.design_building_block_model.Design.equipment.Aviation.Flight_Deck.equipment}
->setup.Equipment_Library.Aviation.flight_deck"
```

```
    Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.Flight_Deck.datum_point.x} = (-" &
Flight_Deck_Info(1); "*(user_variables.LBP))-15[ft]"
    Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.Flight_Deck.datum_point.y} = (-" &
Flight_Deck_Info(2); "*(user_variables.Beam/2))"
    Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.Flight_Deck.datum_point.z} =
(design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations." _
        & Flight_Deck_Info(3); ")+" & Flight_Deck_Info(4); "[ft]"

    If Hanger_Info(11) = "yes" Then

        Print #1, "{setup.Equipment_Library.Aviation} new equipment hanger"
        Print #1, "read dxf (0, 1, 2, 1, 1, 1, 0.000000, 0.000000, 0.000000, 1.000000) " & graphic_path;
"\Hangers\" & Hanger_Info(9); ".dxf"
        Print #1, "{" & Hanger_Info(9); "_dxf.facet_bodies.facet_1} cut"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.visible_geometry} paste"
        Print #1, "{" & Hanger_Info(9); "_dxf} delete"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry} new point
min_x_y_z"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry} new point
max_x_y_z"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry} new solid_body
hanger"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry.min_x_y_z.x} = -" &
Hanger_Info(5) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry.max_x_y_z.x} = " &
Hanger_Info(5) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry.min_x_y_z.y} = -" &
Hanger_Info(6) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry.max_x_y_z.y} = " &
Hanger_Info(6) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry.max_x_y_z.z} = " &
Hanger_Info(7); "[ft]"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry.hanger} cuboid
(setup.Equipment_Library.Aviation" _
            ; ".hanger.construction_geometry.min_x_y_z,
setup.Equipment_Library.Aviation.hanger.construction_geometry.max_x_y_z )"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.construction_geometry.hanger} copy"
        Print #1, "{setup.Equipment_Library.Aviation.hanger.visible_geometry} paste"
        Print #1, "{design.design_building_block_model.Design.equipment.Aviation} new building_block
hangers"
        Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.attributes.use_sub_blocks_i
gnore_this}"
        Print #1, "{design.design_building_block_model.Design.equipment.Aviation.hangers} new
equipment_instance hanger_1"
```

```
        Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.equipment} -
>setup.Equipment_Library.Aviation.hanger"
        Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.datum_point.z} =
(design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations." _
          & Hanger_Info(3); ")+" & Hanger_Info(4); "[ft]"

        If Hanger_Info(8) = 2 Then

          Print #1, "{design.design_building_block_model.Design.equipment.Aviation.hangers} new
equipment_instance hanger_2"
          Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_2.equipment} -
>setup.Equipment_Library.Aviation.hanger"
          Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_2.datum_point.z} =
(design_reference.deck_and_bulkhead_layout.layout_grid.deck_locations." _
            & Hanger_Info(3); ")+" & Hanger_Info(4); "[ft]"

          If Hanger_Info(13) = "side-by-side" Then

            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_2.datum_point.x} = -
(" & _
              Hanger_Info(1); "*user_variables.LBP)"
            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_2.datum_point.y} =
(" & _
              Hanger_Info(2); "*user_variables.Beam/2)+" & ((Hanger_Info(10) / 2) + (Hanger_Info(6) /
2)); "[ft]"
            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.datum_point.x} = -
(" & _
              Hanger_Info(1); "*user_variables.LBP)"
            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.datum_point.y} =
(" & _
              Hanger_Info(2); "*user_variables.Beam/2)-" & ((Hanger_Info(10) / 2) + (Hanger_Info(6) /
2)); "[ft]"

          Else

            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_2.datum_point.x} = -
(" & _
              Hanger_Info(1); "*user_variables.LBP)+" & ((Hanger_Info(10) / 2) + (Hanger_Info(5) / 2));
"[ft]"
```

```
            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_2.datum_point.y} =
(" & _
              Hanger_Info(2); "*user_variables.Beam/2)"
            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.datum_point.x} = -
(" & _
              Hanger_Info(1); "*user_variables.LBP)-" & ((Hanger_Info(10) / 2) + (Hanger_Info(5) / 2));
"[ft]"
            Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.datum_point.y} =
(" & _
              Hanger_Info(2); "*user_variables.Beam/2)"

          End If

        Else

        Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.datum_point.x} = -
(" & _
          Hanger_Info(1); "*user_variables.LBP)"
        Print #1,
"{design.design_building_block_model.Design.equipment.Aviation.hangers.hanger_1.datum_point.y} =
(" & _
          Hanger_Info(2); "*user_variables.Beam/2)"

        End If

      End If

    End If

''''''''''''''''''''''
'Insertion of Equipment'
''''''''''''''''''''''

For n = 1 To 96
    If bb_type((n * 4) - 3) = 0 Then
    Else
      Print #1, "{design.design_building_block_model.Design.equipment} new building_block " &
bb_type((n * 4) - 3)
      Print #1, "{design.design_building_block_model.Design.equipment." & bb_type((n * 4) - 3);
".attributes.use_sub_blocks_ignore_this}"
      Print #1, "{design.design_building_block_model.Design.equipment." & bb_type((n * 4) - 3); "} new
equipment_instance " & bb_name((n * 4) - 3); "_" & n
```

```
      Print #1, "{design.design_building_block_model.Design.equipment." & bb_type((n * 4) - 3); "." &
bb_name((n * 4) - 3); "_" & n; ".equipment} ->setup.Equipment_Library." & bb_type((n * 4) - 3); "." &
bb_name((n * 4) - 3)
      Print #1, "{design.design_building_block_model.Design.equipment." & bb_type((n * 4) - 3); "." &
bb_name((n * 4) - 3); "_" & n; ".datum_point.x} = " & equip_loc_x((n * 4) - 3); "[ft]"
      Print #1, "{design.design_building_block_model.Design.equipment." & bb_type((n * 4) - 3); "." &
bb_name((n * 4) - 3); "_" & n; ".datum_point.y} = " & equip_loc_y((n * 4) - 3); "[ft]"
      Print #1, "{design.design_building_block_model.Design.equipment." & bb_type((n * 4) - 3); "." &
bb_name((n * 4) - 3); "_" & n; ".datum_point.z} = " & equip_loc_z((n * 4) - 3); "[ft]"

   End If

'  Print #1, "new placeholder " & bb_type((n * 4) - 3); "_" & n
 '    Print #1, "{" & bb_type((n * 4) - 3); "_" & n; "} new placeholder " & bb_name((n * 4) - 3); "_" & n

Next n



''''''''''''''''''''''''''''
'Develop MMR Building Blocks'
''''''''''''''''''''''''''''


   Dim MMRs_x, MMRs_y, MMRs_z, Outputs As Range

   Set Outputs = Range("G3:G26")
   MMRs_x = Outputs(6)
   MMRs_y = Outputs(13)
   MMRs_z = Outputs(11)

   Worksheets("MMR").Activate

   Set MMR_equip_name = Range("A8:A612")
   Set MMR_equip_weight = Range("C8:C612")
   Set MMR_equip_weight_x = Range("D8:D612")
   Set MMR_equip_weight_y = Range("E8:E612")
   Set MMR_equip_weight_z = Range("F8:F612")
   Set MMR_equip_length = Range("S8:S612")
   Set MMR_equip_width = Range("T8:T612")
   Set MMR_equip_height = Range("U8:U612")
   Set MMR_equip_x = Range("V8:V612")
   Set MMR_equip_y = Range("W8:W612")
   Set MMR_equip_z = Range("X8:X612")
   Set MMR_equip_room = Range("DY8:DY612")
   Set MMR_equip_SWBS = Range("DW8:DW612")
   Set MMR_equip_graphic = Range("AQ8:AQ612")
```

```
For n = 1 To 604

    Print #1, "{setup.Equipment_Library.Machinery} new equipment " & MMR_equip_name(n)
    Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry} new point min_x_y_z"
        Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry.min_x_y_z.x} = " & -MMR_equip_length(n) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry.min_x_y_z.y} = " & -MMR_equip_width(n) / 2; "[ft]"
    Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry} new point max_x_y_z"
        Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry.max_x_y_z.x} = " & MMR_equip_length(n) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry.max_x_y_z.y} = " & MMR_equip_width(n) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry.max_x_y_z.z} = " & MMR_equip_height(n); "[ft]"
    Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry} new solid_body " & MMR_equip_name(n)
    Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry." & MMR_equip_name(n); _
        "} cuboid (setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry.min_x_y_z, setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry.max_x_y_z)"
    Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n); ".visible_geometry} new
solid_body " & MMR_equip_name(n)

    If MMR_equip_graphic(n) = 0 Then

        Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n); ".visible_geometry." &
MMR_equip_name(n); _
            "} copy_of (setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".construction_geometry." & MMR_equip_name(n); ")"

    Else

        Print #1, "read parasolid (0, 1, 2, 1, 1, 1, 0.000000, 0.000000, 0.000000, 1.000000) " &
graphic_path; "\Machinery\" & MMR_equip_graphic(n); ".x_t"
        Print #1, "{" & MMR_equip_graphic(n); "_x_t} cut"
        Print #1, "{Graphics} paste"
        Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n); ".visible_geometry." &
MMR_equip_name(n); _
            "} copy_of (Graphics." & MMR_equip_graphic(n); ")"

    End If

    Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n); ".characteristics} new
char_weight MMR_equipment_weight"
```

```
      Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".characteristics.MMR_equipment_weight.weight} = " & MMR_equip_weight(n); "[LT]"
      Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".characteristics.MMR_equipment_weight.classification} -> setup.classification_systems." _
          & MMR_equip_SWBS(n)
      Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".characteristics.MMR_equipment_weight.centroid.x_offset} = " & _
          MMR_equip_weight_x(n) - MMR_equip_x(n); "[ft]"
      Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".characteristics.MMR_equipment_weight.centroid.y_offset} = " & _
          MMR_equip_weight_y(n) - MMR_equip_y(n); "[ft]"
      Print #1, "{setup.Equipment_Library.Machinery." & MMR_equip_name(n);
".characteristics.MMR_equipment_weight.centroid.z_offset} = " & _
          MMR_equip_weight_z(n) - MMR_equip_z(n) + (MMR_equip_height(n) / 2); "[ft]"

   Next n

   Worksheets("MMR_Bounding_Boxes").Activate

   Dim OMR_name, OMR_length, OMR_width, OMR_height, OMR_x, OMR_y, OMR_z As Range

   Set OMR_name = Range("A20:A22")
   Set OMR_length = Range("S20:S22")
   Set OMR_width = Range("T20:T22")
   Set OMR_height = Range("U20:U22")
   Set OMR_x = Range("V20:V22")
   Set OMR_y = Range("W20:W22")
   Set OMR_z = Range("X20:X22")

   For n = 1 To 3

      If OMR_name(n) = 0 Then

      Else

         Print #1, "{design.design_building_block_model.Design.MMRs} new building_block " &
OMR_name(n)
         Print #1, "{design.design_building_block_model.Design.MMRs." & OMR_name(n);
".attributes.use_sub_blocks_ignore_this}"
         Print #1, "{design.design_building_block_model.Design.MMRs." & OMR_name(n); "} new
building_block " & OMR_name(n); "_bounding_box"
         Print #1, "{references} new geom_placeholder " & OMR_name(n); "_reference_points"
         Print #1, "{references." & OMR_name(n); "_reference_points} new point max_x_y_z"
         Print #1, "{references." & OMR_name(n); "_reference_points} new point min_x_y_z"
         Print #1, "{references." & OMR_name(n); "_reference_points.max_x_y_z.x} = " & OMR_x(n) +
(OMR_length(n) / 2); "[ft]"
         Print #1, "{references." & OMR_name(n); "_reference_points.max_x_y_z.y} = " & OMR_y(n) +
(OMR_width(n) / 2); "[ft]"
```

```
        Print #1, "{references." & OMR_name(n); "_reference_points.max_x_y_z.z} = " & OMR_z(n) +
(OMR_height(n) / 2); "[ft]"
        Print #1, "{references." & OMR_name(n); "_reference_points.min_x_y_z.x} = " & OMR_x(n) -
(OMR_length(n) / 2); "[ft]"
        Print #1, "{references." & OMR_name(n); "_reference_points.min_x_y_z.y} = " & OMR_y(n) -
(OMR_width(n) / 2); "[ft]"
        Print #1, "{references." & OMR_name(n); "_reference_points.min_x_y_z.z} = " & OMR_z(n) -
(OMR_height(n) / 2); "[ft]"
        Print #1, "{design.design_building_block_model.Design.MMRs." & OMR_name(n); "." &
OMR_name(n); "_bounding_box.attributes.solid} " _
          ; "cuboid (references." & OMR_name(n); "_reference_points.min_x_y_z, references." &
OMR_name(n); "_reference_points.max_x_y_z)"
        Print #1, "{design.design_building_block_model.Design.MMRs." & OMR_name(n); "." & _
          OMR_name(n); "_bounding_box.attributes.solid} Intersect(design.envelope.solid_model.hull)"

      End If

    Next n

    Set room_name = Range("A7:A14")
    Set room_x = Range("V7:V14")
    Set room_y = Range("W7:W14")
    Set room_z = Range("X7:X14")
    Set room_length = Range("S7:S14")
    Set room_width = Range("T7:T14")
    Set room_height = Range("U7:U14")

    For n = 1 To 8

      If room_name(n) = 0 Then
      Else

        Print #1, "{design.design_building_block_model.Design.MMRs} new building_block " &
room_name(n)
          Print #1, "{design.design_building_block_model.Design.MMRs." & room_name(n);
".attributes.use_sub_blocks_ignore_this}"
          Print #1, "{design.design_building_block_model.Design.MMRs." & room_name(n); "} new
building_block " & room_name(n)

        Print #1, "{references} new geom_placeholder " & room_name(n); "_reference_points"
          Print #1, "{references." & room_name(n); "_reference_points} new point max_x_y_z"
          Print #1, "{references." & room_name(n); "_reference_points} new point min_x_y_z"
          Print #1, "{references." & room_name(n); "_reference_points.max_x_y_z.x} = " & room_x(n) +
(room_length(n) / 2) + MMRs_x; "[ft]"
          Print #1, "{references." & room_name(n); "_reference_points.max_x_y_z.y} = " & room_y(n) +
(room_width(n) / 2) + MMRs_y; "[ft]"
          Print #1, "{references." & room_name(n); "_reference_points.max_x_y_z.z} = " & room_z(n) +
(room_height(n) / 2) + MMRs_z; "[ft]"
```

```
            Print #1, "{references." & room_name(n); "_reference_points.min_x_y_z.x} = " & room_x(n) -
(room_length(n) / 2) + MMRs_x; "[ft]"
            Print #1, "{references." & room_name(n); "_reference_points.min_x_y_z.y} = " & room_y(n) -
(room_width(n) / 2) + MMRs_y; "[ft]"
            Print #1, "{references." & room_name(n); "_reference_points.min_x_y_z.z} = " & room_z(n) -
(room_height(n) / 2) + MMRs_z; "[ft]"
            Print #1, "{design.design_building_block_model.Design.MMRs." & room_name(n); "." &
room_name(n); ".attributes.solid} " _
              ; "cuboid (references." & room_name(n); "_reference_points.min_x_y_z, references." &
room_name(n); "_reference_points.max_x_y_z)"
            Print #1, "{design.design_building_block_model.Design.MMRs." & room_name(n); "." & _
              room_name(n); ".attributes.solid} intersect (design.envelope.solid_model.hull)"

        End If

    Next n

    For n = 1 To 604

        If MMR_equip_name(n) = 0 Then
        Else
          Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "} new
equipment_instance " & MMR_equip_name(n); "_" & n
          Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "." &
MMR_equip_name(n); "_" & n; ".equipment} ->setup.Equipment_Library.Machinery." &
MMR_equip_name(n)

          For i = 1 To 8

            If room_name(i) = MMR_equip_room(n) Then

              Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "."
& MMR_equip_name(n); _
                "_" & n; ".datum_point.x} = " & MMRs_x + MMR_equip_x(n); "[ft]"
              Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "."
& MMR_equip_name(n); _
                "_" & n; ".datum_point.y} = " & MMRs_y + MMR_equip_y(n); "[ft]"
              Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "."
& MMR_equip_name(n); _
                "_" & n; ".datum_point.z} = " & MMRs_z + MMR_equip_z(n) - (MMR_equip_height(n) / 2);
"[ft]"

            End If

          Next i

          For i = 1 To 3
```

```
            If OMR_name(i) = MMR_equip_room(n) Then

              Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "."
& MMR_equip_name(n); _
                "_" & n; ".datum_point.x} = " & OMR_x(i) + MMR_equip_x(n); "[ft]"
              Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "."
& MMR_equip_name(n); _
                "_" & n; ".datum_point.y} = " & OMR_y(i) + MMR_equip_y(n); "[ft]"
              Print #1, "{design.design_building_block_model.Design.MMRs." & MMR_equip_room(n); "."
& MMR_equip_name(n); _
                "_" & n; ".datum_point.z} = " & OMR_z(i) + MMR_equip_z(n) - (OMR_height(i) / 2) -
(MMR_equip_height(n) / 2); "[ft]"

            End If

          Next i

        End If

      Next n

''''''''''''''''''''
'Powering Calculations'
''''''''''''''''''''

Print #1, "playback C:\ESSDT\Macros\powering_1.kcl"

Worksheets("Power_Bank").Activate

Set max_prop_diam = Range("G20")
Set max_RPM = Range("G19")

Print #1, "{demonstration_design_1_powering} cut"
Print #1, "{analysis} paste"
Print #1, "{analysis.demonstration_design_1_powering} rename powering"
Print #1, "{analysis.powering.1_Environment.water_density} -> setup.consumable_densities.sea_water"
Print #1, "{analysis.powering.2_analysis_deep.Geometry_Deep_EOL.naked_hull_body} ->
design.envelope.solid_model.hull"

   'developing sheets for powering analysis

            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
profile_point_one"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.profile_point_one.x} =
.25*(user_variables.LBP)"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.profile_point_one.z} = -50[ft]"
```

```
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
profile_point_two"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.profile_point_two.x} = -
1.25*(user_variables.LBP)"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.profile_point_two.z} =
1.25*(user_variables.HOA)"
            Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.sheets.wind_profile_sheet} rectangle
(design_reference.profiles_sheets_clearances_etc.profiles.points.profile_point_one,
design_reference.profiles_sheets_clearances_etc.profiles.points.profile_point_two )"
            Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.sheets.wind_profile_sheet} intersect
(design.envelope.solid_model.hull)"

            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
transverse_point_one"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_one.x} = -
(user_variables.LBP)/2"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_one.y} =
1.1*(user_variables.Beam)"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_one.z} = -50[ft]"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
transverse_point_two"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_two.x} = -
(user_variables.LBP)/2"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_two.y} = -
1.1*(user_variables.Beam)"
              Print #1,
"{design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_two.z} =
1.25*(user_variables.HOA)"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets.transverse_sheet}
rectangle (design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_one,
design_reference.profiles_sheets_clearances_etc.profiles.points.transverse_point_two )"
            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets.transverse_sheet}
intersect (design.envelope.solid_model.hull)"

            Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets.upper_deck_sheet}
copy_of (design.envelope.solid_model.hull_sheets.deck)"

Print #1, "{analysis.powering.2_analysis_deep.Geometry_Deep_EOL.plan_sheet} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.upper_deck_sheet"
```

```
Print #1, "{analysis.powering.2_analysis_deep.Geometry_Deep_EOL.profile_sheet} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.wind_profile_sheet"
Print #1, "{analysis.powering.2_analysis_deep.Geometry_Deep_EOL.transverse_sheet} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.transverse_sheet"
Print #1, "{analysis.powering.2_analysis_deep.Geometry_Deep_EOL.AP_baseline.x} = -
(user_variables.LBP)"
Print #1, "{analysis.powering.2_analysis_deep.Geometry_Deep_EOL.transom_x} = -(user_variables.LBP)"
Print #1, "{analysis.powering.2_analysis_deep.Geometry_Deep_EOL.waterplane.d} =
(user_variables.Draft)"
Print #1, "{analysis.powering.2_analysis_deep.visualiser.speeds.speeds.stop} =
(user_variables.Max_Speed)"
Print #1, "{analysis.powering.2_analysis_deep.propeller_limits.max_diameter} = " & max_prop_diam;
"[ft]"
Print #1, "{analysis.powering.2_analysis_deep.propeller_limits.shaft_speed.max} = " & max_RPM;
"[RPM]"
Print #1, "{analysis.powering.2_analysis_deep.propeller_finder.search_method.neither}"
Print #1, "{analysis.powering.2_analysis_deep.propeller_finder.efficiency_reduction.fixed_pitch}"
Print #1, "{analysis.powering.2_analysis_deep.propeller_finder.design_speed} =
(user_variables.Endurance_Speed)"

''''''''''''''''
'Inserting Stacks'
''''''''''''''''

Worksheets("Output").Activate

  Set Stack_Diameter = Range("J5:J164")
  Set Stack_x = Range("P5:P164")
  Set Stack_y = Range("Q5:Q164")
  Set Stack_z = Range("R5:R164")
  Set zed = Range("S5")
  Set yes = Range("S6")

  Print #1, "{references} new geom_placeholder stacks"

  For n = 1 To 20

    If Stack_Diameter((8 * n) - 7) > 0 Then

      Print #1, "{setup.Equipment_Library.Stacks} new equipment stack_" & n
      Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new var
diameter"
      Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.diameter}
set_units_category (length_default)"
      Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new var
additional_height_of_stack"
      Print #1, "{setup.Equipment_Library.Stacks.stack_" & n;
".construction_geometry.additional_height_of_stack} set_units_category (length_default)"
```

```
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.diameter} = " &
Stack_Diameter((8 * n) - 7); "[ft]"
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n;
".construction_geometry.additional_height_of_stack} = 1[ft]"
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new polyline
stack_polyline"
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new sheet
stack_circle"


        For i = 1 To 8

        If Stack_x((8 * n) - 8 + i) = Stack_x((8 * n) - 9 + i) Then
          If Stack_y((8 * n) - 8 + i) = Stack_y((8 * n) - 9 + i) Then
            If Stack_z((8 * n) - 8 + i) = Stack_z((8 * n) - 9 + i) Then
            Else

              Print #1, "{setup.Equipment_Library.Stacks.stack_" & n;
".construction_geometry.stack_polyline} add_point (" & _
                Stack_x((8 * n) - 8 + i) - Stack_x((8 * n) - 7); "[ft], " & Stack_y((8 * n) - 8 + i) - Stack_y((8 *
n) - 7); "[ft], " _
                & Stack_z((8 * n) - 8 + i) - Stack_z((8 * n) - 7); "[ft])"
            End If
          Else

              Print #1, "{setup.Equipment_Library.Stacks.stack_" & n;
".construction_geometry.stack_polyline} add_point (" & _
                Stack_x((8 * n) - 8 + i) - Stack_x((8 * n) - 7); "[ft], " & Stack_y((8 * n) - 8 + i) - Stack_y((8 *
n) - 7); "[ft], " _
                & Stack_z((8 * n) - 8 + i) - Stack_z((8 * n) - 7); "[ft])"
          End If
        Else

              Print #1, "{setup.Equipment_Library.Stacks.stack_" & n;
".construction_geometry.stack_polyline} add_point (" & _
                Stack_x((8 * n) - 8 + i) - Stack_x((8 * n) - 7); "[ft], " & Stack_y((8 * n) - 8 + i) - Stack_y((8 *
n) - 7); "[ft], " _
                & Stack_z((8 * n) - 8 + i) - Stack_z((8 * n) - 7); "[ft])"

        End If

        Next i

        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.stack_polyline}
add_point (" & _
            Stack_x((8 * n)) - Stack_x((8 * n) - 7); "[ft], " & Stack_y((8 * n)) - Stack_y((8 * n) - 7); "[ft], (" _
            & Stack_z((8 * n)) - Stack_z((8 * n) - 7); "[ft]+setup.Equipment_Library.Stacks.stack_" & n;
".construction_geometry.additional_height_of_stack))"
```

```
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new point
one"
          Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.one.x} = " &
Stack_Diameter((8 * n) - 7) / 2; "[ft]"
          Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.one.y} = " &
Stack_Diameter((8 * n) - 7) / 2; "[ft]"
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new point
two"
          Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.two.x} = " & -
Stack_Diameter((8 * n) - 7) / 2; "[ft]"
          Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.two.y} = " & -
Stack_Diameter((8 * n) - 7) / 2; "[ft]"

        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.stack_circle}
rectangle (setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.one,
setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.two)"

        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new point
center"

        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.stack_circle}
circle (" & zed; ", setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.diameter/2,
setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.center)"

        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry} new wire
stack_wire"
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.stack_wire}
wire_from_polyline (setup.Equipment_Library.Stacks.stack_" & n;
".construction_geometry.stack_polyline)"
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".visible_geometry} new solid_body
stack_" & n
        Print #1, "{setup.Equipment_Library.Stacks.stack_" & n; ".visible_geometry.stack_" & n; "}
sweep_sheet_wire (setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.stack_circle,
setup.Equipment_Library.Stacks.stack_" & n; ".construction_geometry.stack_wire, " & yes; ")"

        Print #1, "{design.design_building_block_model.Design.MMRs.exhaust_stacks} new
equipment_instance stack_" & n
        Print #1, "{design.design_building_block_model.Design.MMRs.exhaust_stacks.stack_" & n;
".equipment} -> setup.Equipment_Library.Stacks.stack_" & n
        Print #1, "{design.design_building_block_model.Design.MMRs.exhaust_stacks.stack_" & n;
".datum_point.x} = " & Stack_x((8 * n) - 7); "[ft]"
        Print #1, "{design.design_building_block_model.Design.MMRs.exhaust_stacks.stack_" & n;
".datum_point.y} = " & Stack_y((8 * n) - 7); "[ft]"
        Print #1, "{design.design_building_block_model.Design.MMRs.exhaust_stacks.stack_" & n;
".datum_point.z} = " & Stack_z((8 * n) - 7); "[ft]"

      End If
```

```
    Next n

''''''''''''''''''''''''''
'Structural Weight Estimate'
''''''''''''''''''''''''''

Worksheets("Output").Activate

Set z_cent_est = Range("C35")

Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.length} = (user_variables.LBP)"
Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.beam} = (user_variables.Beam)"
Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.depth} = (user_variables.Depth_midships)"
Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics} new char_weight weight"
Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.weight.classification} -> setup.classification_systems.Group_1_Structures"
Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.weight.weight} = (0.0035 *
((design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.length [in m]) ^ 2.154) *
(((design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.beam [in m] ^ 2.000) *
design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.charact
eristics.depth [in m]) ^ (1.000 / 3.000)) [te]) * 1"
Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.weight.centroid.x_offset} =
(design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.charac
teristics.x_position)"
Print #1,
"{design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.chara
cteristics.weight.centroid.z_offset} =
(design.design_building_block_model.Design.requirements.structure.overall_estimate.attributes.charac
teristics.z_position)"

''''''''''''''''
'Tankage Estimate'
```

```
''''''''''''''''

Worksheets("Volume_Bank").Activate

Set SFC = Range("I6")
Set electrical_loads = Range("I8:I11")
electrical_load_total = electrical_loads(1) + electrical_loads(2) + electrical_loads(3) + electrical_loads(4)

''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''
''This line appears to not work''done manually
''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''


'Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.propulsion_p
ower_at_endurance_speed} =
(analysis.powering.2_analysis_deep.visualiser.tabular_results.shaft_power.v11)"

Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.electrical_po
wer_at_endurance_speed} = " & electrical_load_total; "[MW]"
Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.time_at_end
urance_speed} = ((user_variables.Range)/(user_variables.Endurance_Speed))"
Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.total_power_
at_endurance_speed} =
((design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.propulsion_p
ower_at_endurance_speed)+(design.design_building_block_model.Design.requirements.fuel.attributes.
characteristics.electrical_power_at_endurance_speed))"
Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.SFC_at_endu
rance_speed} = " & SFC; "[gramme/MWh]"
Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.margins} =
1+(setup.margins.design_margin_multipliers.weight_margin)"
Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.fuel_demand
_for_endurance_speed.consumable_type} -> setup.consumable_densities.dieso_fuel"
Print #1,
"{design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.fuel_demand
_for_endurance_speed.demand} =
(design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.margins *
(design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.SFC_at_endur
ance_speed *
design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.time_at_endur
ance_speed *
design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.total_power_at
_endurance_speed))"


''''''''''''''''''
'Stability Analysis'
''''''''''''''''''


Worksheets("Defaults").Activate

Set margin_line = Range("C70")
Set additional_max_draught = Range("C71")
Set max_trim = Range("C72")
Set fuel_LCG = Range("C73")
Set fuel_VCG = Range("C74")

Worksheets("Output").Activate

Set add_group = Range("A30:A39")
Set add_name = Range("B30:B39")
Set add_weight = Range("C30:C39")
Set add_weight_x = Range("D30:D39")
Set add_weight_y = Range("E30:E39")
Set add_weight_z = Range("F30:F39")

'  Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets.margin_sheet} copy_of
design.envelope.solid_model.hull_deck_sheets.H1"
 '  Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets.margin_sheet} translate
(0[ft], 0[ft], " & -margin_line; "[ft]"

   Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
margin_point_one"
       Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_one.x} =
.25*(user_variables.LBP)"
       Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_one.y} =
1.25*(user_variables.Beam)"
       Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_one.z} =
(user_variables.deck_heights.H1)-" & margin_line; "[ft]"
   Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points} new point
margin_point_two"
       Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_two.x} = -
1.25*(user_variables.LBP)"
       Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_two.y} = -
1.25*(user_variables.Beam)"
       Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_two.z} =
(user_variables.deck_heights.H1)-" & margin_line; "[ft]"
```

```
   Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets.margin_sheet} rectangle
(design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_one,
design_reference.profiles_sheets_clearances_etc.profiles.points.margin_point_two)"
   Print #1, "{design_reference.profiles_sheets_clearances_etc.profiles.sheets.margin_sheet} intersect
(design.envelope.solid_model.hull)"

Print #1, "{analysis.esd_intact_stability.1_criteria} new stab_ass_Def_Stan_02109_shape nes109_shape"
Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.sea_water} =
(setup.consumable_densities.sea_water)"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.AP_baseline.x} = -
(user_variables.LBP)"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.upper_deck_sheet} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.upper_deck_sheet"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.margin_sheet} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.margin_sheet"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.wind_profile} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.wind_profile_sheet"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits} new
attitude_limit deep_limits"
      Print #1,
"{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits.deep_limits.maximum_draug
ht.limiting_value} = (user_variables.Depth_midships)+" & additional_max_draught; "[ft]"
      Print #1,
"{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits.deep_limits.maximum_trim_
BP.limiting_value} = " & max_trim; "[ft]"
      Print #1,
"{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits.deep_limits.condition} ->
setup.ship_conditions.full_load_condition"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits} new
attitude_limit light_limits"
      Print #1,
"{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits.light_limits.maximum_draug
ht.limiting_value} = (user_variables.Depth_midships)+" & additional_max_draught; "[ft]"
      Print #1,
"{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits.light_limits.maximum_trim_B
P.limiting_value} = " & max_trim; "[ft]"
      Print #1,
"{analysis.esd_intact_stability.2_stability_without_margins.attitude_limits.light_limits.condition} ->
setup.ship_conditions.lightship_condition"

   Print #1, "{analysis.esd_intact_stability.1_criteria.nes109_shape} copy"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.template_criteria}
paste_as_pointer concept_criteria_pointer"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.heel_range} new for_next
heel_range"
   Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.heel_range.heel_range.start} = -
50[deg]"
```

```
  Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.heel_range.heel_range.stop} =
50[deg]"
  Print #1,
"{analysis.esd_intact_stability.2_stability_without_margins.heel_range.heel_range.increment} = 5[deg]"
  Print #1, "{analysis.esd_intact_stability.2_stability_without_margins.block_definition} -
>audit.weight.full_load.block_definition_light"

  Print #1, "{audit.discrete_audits.consumables_by_block.block_summary.top_level_block} ->
design.design_building_block_model.Design"
  Print #1, "{audit.discrete_audits.consumables_by_block.block_definition.block_summary} ->
audit.discrete_audits.consumables_by_block.block_summary"
  Print #1, "{audit.discrete_audits.consumables_by_block.design_audit.block_definition} ->
audit.discrete_audits.consumables_by_block.block_definition"
  Print #1, "{audit.discrete_audits.consumables_by_block.design_audit.characteristic.consumables}"
  Print #1, "{audit.discrete_audits.consumables_by_block.design_audit.characteristic.condition} ->
setup.ship_conditions.full_load_condition"

  Print #1, "{design.design_building_block_model.Design.requirements} new building_block
additional_weight_items"
  Print #1,
"{design.design_building_block_model.Design.requirements.additional_weight_items.attributes.use_su
b_blocks_ignore_this}"

  For n = 1 To 10

    Print #1, "{design.design_building_block_model.Design.requirements.additional_weight_items}
new building_block " & add_name(n)
    Print #1, "{design.design_building_block_model.Design.requirements.additional_weight_items." &
_
      add_name(n); ".attributes.characteristics} new char_weight " & add_name(n)
    Print #1, "{design.design_building_block_model.Design.requirements.additional_weight_items." &
_
      add_name(n); ".attributes.characteristics." & add_name(n); ".weight} = " & add_weight(n); "[LT]"
    Print #1, "{design.design_building_block_model.Design.requirements.additional_weight_items." &
_
      add_name(n); ".attributes.characteristics." & add_name(n); ".centroid.x_offset} = " &
add_weight_x(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.requirements.additional_weight_items." &
_
      add_name(n); ".attributes.characteristics." & add_name(n); ".centroid.y_offset} = " &
add_weight_y(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.requirements.additional_weight_items." &
_
      add_name(n); ".attributes.characteristics." & add_name(n); ".centroid.z_offset} = " &
add_weight_z(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.requirements.additional_weight_items." &
_
```

```
        add_name(n); ".attributes.characteristics." & add_name(n); ".classification} ->
setup.classification_systems." & add_group(n)

  Next n

  Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics} new
char_weight_absolute design_margin"
     Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
margin.classification} -> setup.classification_systems.unallocated_weight"
  Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics} new
char_weight_absolute design_allowance"
     Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
alloawance.classification} -> setup.classification_systems.unallocated_weight"

  Print #1, "{audit.weight.simplified_full_load.block_definition_full_load.characteristics} new
char_weight_absolute design_margin"
     Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_margin.classificatio
n} -> setup.classification_systems.unallocated_weight"
  Print #1, "{audit.weight.simplified_full_load.block_definition_full_load.characteristics} new
char_weight_absolute design_allowance"
     Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_alloawance.classific
ation} -> setup.classification_systems.unallocated_weight"
  Print #1, "{audit.weight.simplified_full_load.block_definition_full_load.characteristics} new
char_weight_absolute fuel"
     Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.fuel.classification} ->
setup.classification_systems.Group_8_Variable_Load"


''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''
"This line appears to not work"done manually
''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''


'Print #1, "{audit.weight.margins.lightship} =
(audit.weight.lightship.design_audit.results.1_Total.weight)"

Print #1, "{audit.weight.margins.design_margin} =
((audit.weight.margins.lightship)*(setup.margins.design_margin_multipliers.weight_margin))"
Print #1, "{audit.weight.margins.growth_allowance} =
((audit.weight.margins.lightship)*(setup.margins.design_margin_multipliers.weight_allowance))"
```

```
   Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
margin.weight} = (audit.weight.margins.design_margin)"
   Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
margin.centroid.x} = -((user_variables.LBP)/2)"
   Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
margin.centroid.z} =
((user_variables.deck_heights.Weather_Deck)+(user_variables.deck_heights.H1))/2"
   Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
allowance.weight} = (audit.weight.margins.growth_allowance)"
   Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
allowance.centroid.x} = -((user_variables.LBP)/2)"
   Print #1,
"{audit.weight.lightship_including_margins.block_definition_light_with_margins.characteristics.design_
allowance.centroid.z} = ((user_variables.Depth_midships)/2)"

Print #1, "{audit.weight.simplifications.fuel} =
(design.design_building_block_model.Design.requirements.fuel.attributes.characteristics.fuel_demand_
for_endurance_speed.demand)"
Print #1, "{audit.weight.simplifications.fuel_margin} =
((audit.weight.simplifications.fuel)*((setup.margins.design_margin_multipliers.weight_margin)))"

   Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_margin.weight} =
(audit.weight.margins.design_margin)"
   Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_margin.centroid.x}
= -((user_variables.LBP)/2)"
   Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_margin.centroid.z}
= ((user_variables.deck_heights.Weather_Deck)+(user_variables.deck_heights.H1))/2"
   Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_allowance.weight}
= (audit.weight.margins.growth_allowance)"
   Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_allowance.centroid.
x} = -((user_variables.LBP)/2)"
   Print #1,
"{audit.weight.simplified_full_load.block_definition_full_load.characteristics.design_allowance.centroid.
z} = ((user_variables.Depth_midships)/2)"

   Print #1, "{audit.weight.simplified_full_load.block_definition_full_load.characteristics.fuel.weight} =
(audit.weight.simplifications.fuel)+(audit.weight.simplifications.fuel_margin)"
```

```
   Print #1, "{audit.weight.simplified_full_load.block_definition_full_load.characteristics.fuel.centroid.x}
= -((user_variables.LBP)*" & fuel_LCG; ")"
   Print #1, "{audit.weight.simplified_full_load.block_definition_full_load.characteristics.fuel.centroid.z}
= ((user_variables.Depth_midships)*" & fuel_VCG; ")"

Print #1, "{analysis.stability} new stab_placeholder stability_analysis"
   Print #1, "{analysis.stability.stability_analysis} new hull_envelope hull_envelope"
   Print #1, "{analysis.stability.stability_analysis.hull_envelope.buoyant_bodies} new body_pointer hull"
   Print #1, "{analysis.stability.stability_analysis.hull_envelope.buoyant_bodies.hull} ->
design.envelope.solid_model.hull"
   Print #1, "{analysis.stability.stability_analysis.hull_envelope.upper_deck_sheet} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.upper_deck_sheet"
   Print #1, "{analysis.stability.stability_analysis.hull_envelope.margin_sheet} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.margin_sheet"
   Print #1, "{analysis.stability.stability_analysis.hull_envelope.wind_profile} ->
design_reference.profiles_sheets_clearances_etc.profiles.sheets.wind_profile_sheet"
   Print #1, "{analysis.stability.stability_analysis.hull_envelope.AP_baseline.x} = -(user_variables.LBP)"
   Print #1, "{analysis.stability.stability_analysis.hull_envelope.AM_baseline.x} = -(user_variables.LBP)"
   Print #1, "{analysis.stability.stability_analysis} new stab_placeholder densities"
   Print #1, "{analysis.stability.stability_analysis.densities} new density sea_water"
   Print #1, "{analysis.stability.stability_analysis.densities} new density dieso_fuel"
   Print #1, "{analysis.stability.stability_analysis.densities} new density fresh_water"
   Print #1, "{analysis.stability.stability_analysis.densities} new density lube_oil"
      Print #1, "{analysis.stability.stability_analysis.densities.sea_water} =
(setup.consumable_densities.sea_water)"
      Print #1, "{analysis.stability.stability_analysis.densities.dieso_fuel} =
(setup.consumable_densities.dieso_fuel)"
      Print #1, "{analysis.stability.stability_analysis.densities.fresh_water} =
(setup.consumable_densities.fresh_water)"
      Print #1, "{analysis.stability.stability_analysis.densities.lube_oil} =
(setup.consumable_densities.lube_oil)"
   Print #1, "{analysis.stability.stability_analysis} new stab_settings stability_settings"
   Print #1, "{analysis.stability.stability_analysis} new basic_ship simplified_full_load"
      Print #1, "{analysis.stability.stability_analysis.simplified_full_load.stability_settings} ->
analysis.stability.stability_analysis.stability_settings"
      Print #1, "{analysis.stability.stability_analysis.simplified_full_load.hull_envelope} ->
analysis.stability.stability_analysis.hull_envelope"

''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''
"These lines appears to not work"done manually
''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''

'   Print #1, "{analysis.stability.stability_analysis.simplified_full_load.datum.weight} =
(audit.weight.lightship_including_margins.design_audit.results.1_Total.weight)"
'   Print #1, "{analysis.stability.stability_analysis.simplified_full_load.datum.centroid_x} =
(audit.weight.lightship_including_margins.design_audit.results.1_Total.centroid_x)"
```

```
  ' Print #1, "{analysis.stability.stability_analysis.simplified_full_load.datum.centroid_y} =
(audit.weight.lightship_including_margins.design_audit.results.1_Total.centroid_y)"
  ' Print #1, "{analysis.stability.stability_analysis.simplified_full_load.datum.centroid_z} =
(audit.weight.lightship_including_margins.design_audit.results.1_Total.centroid_z)"


   Print #1, "{analysis.stability.stability_analysis.simplified_full_load.water_density} ->
analysis.stability.stability_analysis.densities.sea_water"
   Print #1, "{analysis.stability.stability_analysis} new stab_placeholder loading_conditions"
      Print #1, "{analysis.stability.stability_analysis.loading_conditions} new loading_condition
simplified_full_load"
         Print #1, "{analysis.stability.stability_analysis.loading_conditions.simplified_full_load.basic_ship} -
> analysis.stability.stability_analysis.simplified_full_load"
         Print #1,
"{analysis.stability.stability_analysis.loading_conditions.simplified_full_load.water_density} ->
analysis.stability.stability_analysis.densities.sea_water"

   Print #1, "{analysis.stability.stability_analysis} new stab_placeholder GZ_curves"
      Print #1, "{analysis.stability.stability_analysis.GZ_curves} new GZ GZ_simplified_full_load"
         Print #1,
"{analysis.stability.stability_analysis.GZ_curves.GZ_simplified_full_load.stability_settings} ->
analysis.stability.stability_analysis.stability_settings"
         Print #1, "{analysis.stability.stability_analysis.GZ_curves.GZ_simplified_full_load.load_condition} -
> analysis.stability.stability_analysis.loading_conditions.simplified_full_load"
         Print #1, "{analysis.stability.stability_analysis.GZ_curves.GZ_simplified_full_load.heel_range} new
for_next range"
         Print #1,
"{analysis.stability.stability_analysis.GZ_curves.GZ_simplified_full_load.heel_range.range.start} = -
70[deg]"
         Print #1,
"{analysis.stability.stability_analysis.GZ_curves.GZ_simplified_full_load.heel_range.range.stop} =
70[deg]"
         Print #1,
"{analysis.stability.stability_analysis.GZ_curves.GZ_simplified_full_load.heel_range.range.increment} =
5[deg]"

'cleaning up superfluous data

Print #1, "{setup.Equipment_Library.Machinery.0} delete"

For n = 1 To 1000

   Print #1, "{setup.Equipment_Library." & bb_type(n); "." & bb_name(n);
".visible_geometry.clearance_box_" & (n + 3); "} delete"

Next n

Worksheets("Import_to_Paramarine").Activate
```

```
Application.Calculation = xlManual

   Call ExportKCL(output_file)
End Sub
```

Code for Deckhouse Insertion

```
Sub create_deckhouse()

  Call find_files
    'find_files is in Generic Functions
    Call open_output_file(output_file)

  Application.Calculation = xlAutomatic

  Application.Calculation = xlManual

  Worksheets("Deckhouse_Input").Activate

  'shaping deckhouse to hull

  Print #1, "playback C:\ESSDT\Macros\superstructure_1.kcl"

  Set master_angle = Range("Q7")

    Print #1, "{design_reference.dimensions.superstructure_dimensions.angle_on_sides} = " &
master_angle; "[deg]"
    Print #1, "{design_reference.dimensions.superstructure_dimensions.angle_on_ends} = " &
master_angle; "[deg]"
    Print #1, "{demonstration_file_superstructure_modelling} cut"
    Print #1, "{design_reference.dimensions} paste"
    Print #1, "{design_reference.dimensions.demonstration_file_superstructure_modelling} rename
superstructure_modeling"
    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.port.angle_on_sides}
= (design_reference.dimensions.superstructure_dimensions.angle_on_sides)"
    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.stbd.angle_on_sides}
= (design_reference.dimensions.superstructure_dimensions.angle_on_sides)"
    Print #1, "{design_reference.dimentions.superstructure_modeling.hull_sides.stdb.sweep_height} =
100[ft]"
    Print #1, "{design_reference.dimentions.superstructure_modeling.hull_sides.port.sweep_height} =
100[ft]"
    Print #1,
"{design_reference.dimensions.superstructure_modeling.hull_sides.port.hull_edges.sheet_or_solid} -
>design.envelope.solid_model.hull_sheets.port_hull"
    Print #1,
"{design_reference.dimensions.superstructure_modeling.hull_sides.stbd.hull_edges.sheet_or_solid} -
>design.envelope.solid_model.hull_sheets.stbd_hull"
    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.port.deck_edge}
copy_of
(design_reference.dimensions.superstructure_modeling.hull_sides.port.hull_edges.boundary_edges.bo
undary_4)"
    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.stbd.deck_edge}
copy_of
```

```
(design_reference.dimensions.superstructure_modeling.hull_sides.stbd.hull_edges.boundary_edges.bo
undary_4)"


  'inserting deckhouse blocks

    Set num_mod = Range("Q5")
    Set name_mod = Range("AR22:AR45")
    Set X_fwd = Range("AK22:AK45")
    Set y_port = Range("AL22:AL45")
    Set z_btm = Range("AM22:AM45")
    Set X_aft = Range("AN22:AN45")
    Set y_stbd = Range("AO22:AO45")
    Set z_top = Range("AP22:AP45")
    Set mod_type = Range("AF22:AF45")
    Set theta = Range("AS22:AS45")
    Set zed = Range("AT22:AT45")

    Print #1, "{references} new geom_placeholder deckhouse_cutting_blocks"

'     Print #1, "{references} new geom_placeholder united_deckhouse"
 '    Print #1, "{references.united_deckhouse} new point deck_one"
  '      Print #1, "{references.united_deckhouse.deck_one.y} = .1[ft]"
   '     Print #1, "{references.united_deckhouse.deck_one.z} = .1[ft]+" & z_btm(1); "[ft]"
    ' Print #1, "{references.united_deckhouse} new point deck_two"
     '   Print #1, "{references.united_deckhouse.deck_two.y} = -.1[ft]"
      '  Print #1, "{references.united_deckhouse.deck_two.z} = " & z_btm(1); "[ft]"
       ' Print #1, "{references.united_deckhouse.deck_two.x} = -.75*(user_variables.LBP)"
    'Print #1, "{design.envelope.solid_model.superstructure} cuboid
(references.united_deckhouse.deck_two, references.united_deckhouse.deck_one)"

  For n = 1 To num_mod
    Print #1, "{setup.blocks_to_copy_from.standard_blocks.generic_blank_block_type_1} copy"
    Print #1, "{design.design_building_block_model.Design.deckhouse} paste"
    Print #1, "{design.design_building_block_model.Design.deckhouse.generic_blank_block_type_1}
rename "; name_mod(n)
    Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.X_fwd} = "; X_fwd(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.X_aft} = "; X_aft(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.Y_port} = "; y_port(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.Y_stbd} = "; y_stbd(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.Z_top} = "; z_top(n); "[ft]"
    Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.Z_btm} = "; z_btm(n); "[ft]"
```

```
      Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.solid} cuboid_from_extents ({design.design_building_block_model.Design.deckhouse.";
name_mod(n); ".attributes.characteristics.X_size,
{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.Y_size, {design.design_building_block_model.Design.deckhouse.";
name_mod(n); ".attributes.characteristics.Z_size )"

      Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.solid} translate ((design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.X_fwd+design.design_building_block_model.Design.deckhouse.";
name_mod(n); ".attributes.characteristics.X_aft)/2,
(design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.Y_port+design.design_building_block_model.Design.deckhouse.";
name_mod(n); ".attributes.characteristics.Y_stbd)/2,
(design.design_building_block_model.Design.deckhouse."; name_mod(n);
".attributes.characteristics.Z_top+design.design_building_block_model.Design.deckhouse.";
name_mod(n); ".attributes.characteristics.Z_btm)/2 )"

      theta(n) = Atn((X_fwd(n) - X_aft(n)) / (y_port(n) - y_stbd(n)))

      If mod_type(n) = "triangle_type_1" Then

         Print #1, "{references.deckhouse_cutting_blocks} new geom_placeholder " & name_mod(n)
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point one"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point two"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new solid_body " &
name_mod(n)
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.x} = " & X_aft(n); "[ft]"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.x} = 0[ft]"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.y} = " & y_port(n);
"[ft]"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.y} = " & y_port(n) - (10
* (y_port(n) - y_stbd(n))); "[ft]"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.z} = " & z_top(n) + 1;
"[ft]"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.z} = " & z_btm(n) - 1;
"[ft]"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
            "} cuboid (references.deckhouse_cutting_blocks." & name_mod(n); ".one,
references.deckhouse_cutting_blocks." & name_mod(n); ".two)"
         Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
            "} rotate (" & zed(n); ", " & theta(n); "[rad], references.deckhouse_cutting_blocks." &
name_mod(n); ".one)"
         Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n); "} subtract
(references.deckhouse_cutting_blocks." & _
            name_mod(n); "." & name_mod(n); ")"
```

```
    ElseIf mod_type(n) = "triangle_type_2" Then

        Print #1, "{references.deckhouse_cutting_blocks} new geom_placeholder " & name_mod(n)
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point one"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point two"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new solid_body " &
name_mod(n)
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.x} = " & X_aft(n); "[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.x} = 0[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.y} = " & y_stbd(n);
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.y} = " & y_stbd(n) + (10
* (y_port(n) - y_stbd(n))); "[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.z} = " & z_top(n) + 1;
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.z} = " & z_btm(n) - 1;
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
          "} cuboid (references.deckhouse_cutting_blocks." & name_mod(n); ".one,
references.deckhouse_cutting_blocks." & name_mod(n); ".two)"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
          "} rotate (" & zed(n); ", " & -theta(n); "[rad], references.deckhouse_cutting_blocks." &
name_mod(n); ".one)"
        Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n); "} subtract
(references.deckhouse_cutting_blocks." & _
          name_mod(n); "." & name_mod(n); ")"

    ElseIf mod_type(n) = "triangle_type_3" Then

        Print #1, "{references.deckhouse_cutting_blocks} new geom_placeholder " & name_mod(n)
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point one"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point two"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new solid_body " &
name_mod(n)
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.x} = " & X_fwd(n); "[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.x} = " & 2 * X_aft(n);
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.y} = " & y_stbd(n);
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.y} = " & y_stbd(n) + (10
* (y_port(n) - y_stbd(n))); "[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.z} = " & z_top(n) + 1;
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.z} = " & z_btm(n) - 1;
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
          "} cuboid (references.deckhouse_cutting_blocks." & name_mod(n); ".one,
references.deckhouse_cutting_blocks." & name_mod(n); ".two)"
```

```
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
          "} rotate (" & zed(n); ", " & theta(n); "[rad], references.deckhouse_cutting_blocks." &
name_mod(n); ".one)"
        Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n); "} subtract
(references.deckhouse_cutting_blocks." & _
          name_mod(n); "." & name_mod(n); ")"

      ElseIf mod_type(n) = "triangle_type_4" Then

        Print #1, "{references.deckhouse_cutting_blocks} new geom_placeholder " & name_mod(n)
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point one"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new point two"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "} new solid_body " &
name_mod(n)
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.x} = " & X_fwd(n); "[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.x} = " & 2 * X_aft(n);
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.y} = " & y_port(n);
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.y} = " & y_port(n) - (10
* (y_port(n) - y_stbd(n))); "[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".one.z} = " & z_top(n) + 1;
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); ".two.z} = " & z_btm(n) - 1;
"[ft]"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
          "} cuboid (references.deckhouse_cutting_blocks." & name_mod(n); ".one,
references.deckhouse_cutting_blocks." & name_mod(n); ".two)"
        Print #1, "{references.deckhouse_cutting_blocks." & name_mod(n); "." & name_mod(n); _
          "} rotate (" & zed(n); ", " & -theta(n); "[rad], references.deckhouse_cutting_blocks." &
name_mod(n); ".one)"
        Print #1, "{design.design_building_block_model.Design.deckhouse."; name_mod(n); "} subtract
(references.deckhouse_cutting_blocks." & _
          name_mod(n); "." & name_mod(n); ")"

      End If

    Print #1, "{analysis.stability.stability_analysis.hull_envelope.buoyant_bodies} new body_pointer " &
name_mod(n)
    Print #1, "{analysis.stability.stability_analysis.hull_envelope.buoyant_bodies." & _
      name_mod(n); "} -> design.design_building_block_model.Design.deckhouse." & name_mod(n);
".attributes.solid"


    Next n

    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.port.sweep_height} =
100[ft]"
```

```
    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.stbd.sweep_height} =
100[ft]"

    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.port.deck_edge}
copy_of
(design_reference.dimensions.superstructure_modeling.hull_sides.port.hull_edges.boundary_edges.bo
undary_4)"
    Print #1, "{design_reference.dimensions.superstructure_modeling.hull_sides.stbd.deck_edge}
copy_of
(design_reference.dimensions.superstructure_modeling.hull_sides.stbd.hull_edges.boundary_edges.bo
undary_4)"

    Worksheets("Import_to_Paramarine").Activate

    Application.Calculation = xlAutomatic

    Application.Calculation = xlManual

    Call ExportKCL(output_file)
End Sub
```

Code for Excel to Paramarine Interface

```
Option Explicit

Public Type ParamarineObject
  type As String
  name As String
  location As String
  Pointer1 As String
  Pointer2 As String
End Type

Public MessengerPath As String
Public KCLPath As String
Public input_file As String
Public output_file As String

Function open_output_file(output_file)
  Open output_file For Output As #1
  'turn off error messages
  Print #1, "%error_off"
End Function

Function open_input_file()
  input_file = KCLPath & "\" & "KCLout.kcl"
  Open input_file For Input As #2
End Function

Function ExportKCL(output_file) As Boolean  'function to call ParaMessenger and execute output_file
  'turn on error messages C:\Program Files (x86)\Graphics Research Corporation\Paramarine V7.0\bin
  Print #1, "%error_on"
  Close #1
  Shell (MessengerPath + "\ParaMessenger.exe \d " + output_file)
End Function

Function Wait() 'function to wait for 2 seconds in order to give Paramarine time to write the KCL
  Dim newHour, newMinute, newSecond, waitTime As Double
  newHour = Hour(Now())
  newMinute = Minute(Now())
  newSecond = Second(Now()) + 2
  waitTime = TimeSerial(newHour, newMinute, newSecond)
  Application.Wait waitTime
End Function

Function find_files()
  MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics Research
Corporation\Paramarine7.0", "bin")
  If MessengerPath = "" Then MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics
Research Corporation\Paramarine6.0", "bin")
```

```
  If MessengerPath = "" Then MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics
Research Corporation\Paramarine6.0 Beta", "bin")
  If MessengerPath = "" Then MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics
Research Corporation\Paramarine5.1 Beta", "bin")

  KCLPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics Research Corporation\Application
Profiles\Paramarine\FilePaths", "KCLPath")
  input_file = KCLPath & "\" & "KCLout.kcl"
  output_file = KCLPath & "\" & "KCLin.kcl"
End Function

Sub check_file()
  Call find_files
  Set Fs = CreateObject("scripting.filesystemobject")
  If Fs.fileexists(input_file) Then
    Call next_step
  Else
    Call start_time
  End If
End Sub

Function remove_pipes(input_line As String) 'function to remove pipes from kcl files
  Dim old_line As String
  If InStr(input_line, "|") > 0 Then  'remove pipes
    old_line = Left(input_line, InStr(input_line, "|") - 1)
    Line Input #2, input_line
    If InStr(input_line, "|") > 0 Then
      Call remove_pipes(input_line)
    End If
    input_line = old_line + input_line
  End If
End Function

Function GetObjects(spec_type, object_num, object() As ParamarineObject) As Boolean 'function to load
Paramarine objects into an array
  'spec_type = Paramarine object type to import
  'object_num = number of object imported
  Dim input_line As String
  Dim a As Integer
  ReDim object(9999)
  a = 0
  GetObjects = False 'Boolean output for error handling

  Open input_file For Input As #2    ' Open file for input.
    Line Input #2, input_line
    Do While Not EOF(2)    ' Loop until end of file.
      Line Input #2, input_line
      Call remove_pipes(input_line) 'remove pipes from kcl line
```

```
        If InStr(input_line, "new") > 0 Then
            'if kcl line has Paramarine object type then extract object name and location and type
            If InStr(input_line, "new " & spec_type) > 0 Then
                GetObjects = 1 'Boolean output for error handling
                object(a).location = Mid(input_line, 2, InStr(input_line, "}") - 2)
                object(a).type = Mid(input_line, InStr(input_line, "new") + 4, InStrRev(input_line, " ") -
InStr(input_line, "new") - 4)
                object(a).name = Trim(Mid(input_line, InStr(input_line, " " & object(a).type & " ") +
Len(object(a).type) + 2, Len(input_line)))
                a = a + 1
            End If
        End If
    Loop
  Close #2

  If a <= 0 Then
    GetObjects = False
  Else
    GetObjects = True
    object_num = a - 1
    ReDim Preserve object(object_num)
  End If
End Function

Function get_something(msg As String, obj_type As String) As ParamarineObject()
  Dim para_obj() As ParamarineObject
  Dim object_num As Integer
  MsgBox "Select " & msg
  open_output_file (output_file)
  Print #1, "write kcl " & KCLPath & "\KCLOut.kcl"
  Call ExportKCL(output_file)
  Call Wait

  If GetObjects(obj_type, object_num, para_obj) = False Then
    MsgBox "No " & obj_type & " objects have been found!"
    ReDim para_obj(0)
  End If
  Close #2
  get_something = para_obj
End Function

Function GetFaces(object_num, object() As ParamarineObject) As Boolean 'function to load Paramarine
objects into an array
  'spec_type = Paramarine object type to import
  'object_num = number of object imported
  Dim input_line As String
  Dim temp_str As String
  Dim a As Integer
  ReDim object(9999)
  a = 0
  GetFaces = False 'Boolean output for error handling

  Open input_file For Input As #2   ' Open file for input.
    Do While Not EOF(2)   ' Loop until end of file.
      Line Input #2, input_line
      Call remove_pipes(input_line) 'remove pipes from kcl line
      If InStr(input_line, ".f") > 0 Then
        'if kcl line has Paramarine object type then extract object name and location and type
        GetFaces = 1 'Boolean output for error handling
        temp_str = Mid(input_line, 2, InStr(input_line, "}") - 2)
        object(a).location = Left(temp_str, InStrRev(temp_str, ".") - 1)
        object(a).name = Right(temp_str, Len(temp_str) - InStrRev(temp_str, "."))
        object(a).type = "face"
        object(a).Pointer1 = Right(input_line, Len(input_line) - InStrRev(input_line, "->") - 2)
        a = a + 1
      End If
    Loop
  Close #2

  If a <= 0 Then
    GetFaces = False
  Else
    GetFaces = True
    object_num = a - 1
    ReDim Preserve object(object_num)
  End If
End Function

Function find_pointer(object_num, object() As ParamarineObject, pointer() As ParamarineObject,
num_pointer, pointer_type, pointer_no) 'function to find specific pointers within an array of objects
  'object_num = number of array object to analyze
  'object = Paramarine object array as input array
  'pointer = Paramarine abject array as output array
  'num_pointer = number of array object in output array
  'pointer_type = type of Paramarine pointer eg. "GZ_curve"

  Dim initial_pointer(9999)
  Dim input_line As String
  ReDim pointer(9999)
  Open output_file For Output As #1 'open file for output
  Print #1, "%error_off" 'turn_off_messages
  Print #1, "deselect all"
  For b = 0 To object_num
    Print #1, "select " + object(b).location + "." + object(b).name + "." + pointer_type 'print to file all
object to analyze
  Next b
```

```
    Print #1, "osoakcl " & input_file
    Call ExportKCL(output_file)
    Call Wait
    For a = 0 To object_num
        Open input_file For Input As #2    ' Open file for input.
        Do While Not EOF(2)    ' Loop until end of file.
            Line Input #2, input_line
            Call remove_pipes(input_line) 'remove pipes from current kcl line
            'if kcl line has Paramarine object name and pointer in then extract pointer name and location
            If InStr(input_line, "." + object(a).name + "." + pointer_type) > 0 Then
                initial_pointer(a) = Mid(input_line, InStr(input_line, "." + object(a).name + "." + pointer_type) +
Len("." + object(a).name + "." + pointer_type) + 5)
                    If initial_pointer(a) = "(null)" Then
                    Select Case pointer_type
                        Case "damage_case" 'ok to let null through
                            Select Case pointer_no
                            Case 1
                                object(a).Pointer1 = "null"
                            Case 2
                                object(a).Pointer2 = "null"
                        End Select
                        Case Else
                            MsgBox ("One of your selected criteria is NOT VALID")
                            End
                    End Select
                    Else
                        Select Case pointer_no
                            Case 1
                                object(a).Pointer1 = Right(input_line, Len(input_line) - InStrRev(input_line, "."))
                            Case 2
                                object(a).Pointer2 = Right(input_line, Len(input_line) - InStrRev(input_line, "."))
                        End Select
                    End If
                Exit Do
            End If
        Loop
    Close #2
    Next a
    If a > object_num + 1 Then
        MsgBox "Could not find " + pointer_type
        End
    End If

    If object_num > 0 Then
        Call FilterDuplicates(initial_pointer) 'remove duplicate entries
    End If

    For a = 0 To UBound(initial_pointer) 'remove empty array entries
    If IsEmpty(initial_pointer(a)) Then Exit For
  Next a
  num_pointer = a - 1

  For b = LBound(initial_pointer) To num_pointer 'input pointer name and location into Paramarine
object array
    If initial_pointer(b) <> "(null)" Then
      pointer(b).name = Right(initial_pointer(b), Len(initial_pointer(b)) - InStrRev(initial_pointer(b), "."))
      pointer(b).location = Left(initial_pointer(b), InStrRev(initial_pointer(b), ".") - 1)
    Else
      pointer(b).name = "null"
      pointer(b).location = "null"
    End If
    pointer(b).type = pointer_type
  Next b

  ReDim Preserve pointer(num_pointer)
End Function

Sub start_time()
  newTime = Now + TimeSerial(0, 0, 1)
  Application.OnTime newTime, "check_file"
End Sub

Sub ValvesOff()
  Call find_files
  Open output_file For Output As #1
    Print #1, "%xl_input_close" 'turn_off_valves
    Print #1, "%xl_output_close" 'turn_off_valves
  Close #1
  Call ExportKCL(output_file)
End Sub

Sub ValvesOn()
  Call find_files
  Open output_file For Output As #1
    Print #1, "%xl_input_open" 'turn_on_valves
    Print #1, "%xl_output_open" 'turn_on_valves
  Close #1
  Call ExportKCL(output_file)
End Sub

Sub MsgOff()
  Call find_files
  Open output_file For Output As #1
    Print #1, "%error_off" 'turn_off_messages
  Close #1
  Call ExportKCL(output_file)
```

```
End Sub

Sub MsgOn()
   Call find_files
   Open output_file For Output As #1
      Print #1, "%error_on" 'turn_on_messages
   Close #1
   Call ExportKCL(output_file)
End Sub

Function valid_kcl(validity_check) As Boolean
   valid_kcl = 1
   Open input_file For Input As #2    ' Open file for input.
      Do While Not EOF(2)
         Input #2, input_line
         If InStr(input_line, validity_check) > 0 Then
            valid_kcl = 0
         End If
      Loop
   Close #2
End Function

Function FilterDuplicates(arr As Variant) As Long 'function to remove duplicates from arrays
   Dim col As Collection, index As Long, dups As Long
   Set col = New Collection
   On Error Resume Next
   For index = LBound(arr) To UBound(arr)
      ' build the key using the array element
      ' an error occurs if the key already exists
      col.Add 0, CStr(arr(index))
      If Err Then
         ' we've found a duplicate
         arr(index) = Empty
         dups = dups + 1
         Err.Clear
      ElseIf dups Then
         ' if we've found one or more duplicates so far
         ' we need to move elements towards lower indices
         arr(index - dups) = arr(index)
         arr(index) = Empty
      End If
   Next

   ' return the number of duplicates
   FilterDuplicates = dups
End Function
```

```
Function find_buoyant_bodies(object_num, object() As ParamarineObject, pointer() As
ParamarineObject, num_pointer) 'function to find buoyant bodies within hull envelope object
  'object_num = number of array object to analyze
  'object = Paramarine object array as input array
  'pointer = Paramarine abject array as output array
  'num_pointer = number of array object in output array

  Dim solid_body_pointer(99)
  Dim input_line As String
  ReDim pointer(99)
  Open output_file For Output As #1
    Print #1, "%error_off" 'turn_off_messages
    For b = 0 To object_num
      Print #1, "Deselect all"
      Print #1, "select " + object(b).location + "." + object(b).name
      Print #1, "write kcl " & KCLPath & "\KCLout.kcl"
    Next b
  Call ExportKCL(output_file)
  Call Wait

  Open input_file For Input As #2   ' Open file for input.
  b = 0
  Do While Not EOF(2)   ' Loop until end of file.
      Input #2, input_line
      Call remove_pipes(input_line)
      If InStr(input_line, ".buoyant_bodies.") > 0 Then
        pointer(b).name = Mid(input_line, InStrRev(input_line, ".") + 1)
        pointer(b).location = Mid(input_line, InStr(input_line, "->") + 2, Len(input_line) -
InStr(input_line, "->") - Len(pointer(b).name) - 2)
        solid_body_pointer(b) = Left(input_line, InStr(input_line, "->") - 2)
        pointer(b).Pointer1 = Mid(solid_body_pointer(b), InStrRev(solid_body_pointer(b), ".") + 1)
        pointer(b).type = "buoyant_bodies"
        b = b + 1
      End If
  Loop
  num_pointer = b - 1
  Close #2
  ReDim Preserve pointer(num_pointer)
End Function

Function clean_names() 'function to delecte any labels within Excel which have been left over from
previous operations
  endnum = ActiveWorkbook.names.Count - 1
  a = 1
  Do While a < endnum
    On Error Resume Next
    If InStr(ActiveWorkbook.names(a).value, "REF") > 0 Then 'remove Excel lable if it contains "REF"
      ActiveWorkbook.names(a).Delete
```

```
        a = a - 1
      End If
      a = a + 1
      endnum = ActiveWorkbook.names.Count - 1
    Loop
End Function

Function names()
    'MsgBox Range("NES109_wind_angle_of_heel_1").Address
    criteriaangle_of_heel = "criteria.angle_of_heel"
    crit_col = "crit_col"
    object_path = "object_path"
    criteriaPointer1 = "criteria(b).Pointer1"
    criteria_col = "1"
    'MsgBox "send_this_to_excel(""" & criteriaPointer1 & Range("NES109_wind_angle_of_heel_" &
criteria_col & "").Address & """)"
    'msgbox "{" + object_path + ".criteriaangle_of_heel." & crit_col & "}send_this_to_excel(""" &
criteria(bPointer1 & Range("NES109_wind_angle_of_heel_" & criteria_col).Address & """)"
End Function

Function create_plane(name, x, y, z, D)
    Print #1, "deselect all"
    Print #1, "select structures.geom"
    Print #1, "new plane " & name
    Print #1, "deselect all"
    Print #1, "select structures.geom." & name & ".x"
    Print #1, "=" & x
    Print #1, "deselect all"
    Print #1, "select structures.geom." & name & ".y"
    Print #1, "=" & y
    Print #1, "deselect all"
    Print #1, "select structures.geom." & name & ".z"
    Print #1, "=" & z
    Print #1, "Deselect all"
    Print #1, "select structures.geom." & name & ".d"
    Print #1, "=" & D & "[m]"
End Function

Function get_value(objects() As ParamarineObject, value, value_name As String)
    Dim i As Integer
    Dim input_line As String
    Dim Temp As String
    ReDim value(9999)
    open_output_file (output_file)

    Print #1, "deselect all"
    For i = 0 To UBound(objects)
      Print #1, "select " & objects(i).location & "." & objects(i).name
```

```
  Next i
  Print #1, "osoakcl " + QueryValue(HKEY_CURRENT_USER, "Software\Graphics Research
Corporation\Application Profiles\Paramarine\FilePaths", "KCLPath") + "\KCLOut.kcl"
  Call ExportKCL(output_file)
  Call Wait

  i = 0
  Open input_file For Input As #2   ' Open file for input.
    Input #2, input_line
    Do While Not EOF(2)   ' Loop until end of file.
      Input #2, input_line
      Call remove_pipes(input_line) 'remove pipes from kcl line
      If InStr(input_line, value_name) > 0 Then
        'if kcl line has Paramarine object type then extract object name and location and type
        Temp = Right(input_line, Len(input_line) - InStr(input_line, " = ") - 2)
        Temp = Left(Temp, InStr(Temp, " [") - 2)
        value(i) = CDbl(Temp)
        i = i + 1
      End If
    Loop
  Close #2
  ReDim Preserve value(i - 1)
End Function

' Heap sort routine.
' Returns a sorted Index array for the Keys array.
' Author: Christian d'Heureuse (www.source-code.biz)
Public Function HeapSort(Keys)
  Dim Base As Long: Base = LBound(Keys)               ' array index base
  Dim n As Long: n = UBound(Keys) - LBound(Keys) + 1     ' array size
  ReDim index(Base To Base + n - 1) As Long            ' allocate index array
  Dim i As Long, m As Long
  For i = 0 To n - 1: index(Base + i) = Base + i: Next   ' fill index array
  For i = n \ 2 - 1 To 0 Step -1                       ' generate ordered heap
    Heapify Keys, index, i, n
    Next
  For m = n To 2 Step -1
    Exchange index, 0, m - 1                           ' move highest element to top
    Heapify Keys, index, 0, m - 1
    Next
  HeapSort = index
End Function

Private Sub Heapify(Keys, index() As Long, ByVal i1 As Long, ByVal n As Long)
  ' Heap order rule: a[i] >= a[2*i+1] and a[i] >= a[2*i+2]
  Dim Base As Long: Base = LBound(index)
  Dim nDiv2 As Long: nDiv2 = n \ 2
  Dim i As Long: i = i1
  Do While i < nDiv2
    Dim k As Long: k = 2 * i + 1
    If k + 1 < n Then
      If Keys(index(Base + k)) < Keys(index(Base + k + 1)) Then k = k + 1
      End If
    If Keys(index(Base + i)) >= Keys(index(Base + k)) Then Exit Do
    Exchange index, i, k
    i = k
    Loop
End Sub

Private Sub Exchange(a() As Long, ByVal i As Long, ByVal j As Long)
  Dim Base As Long: Base = LBound(a)
  Dim Temp As Long: Temp = a(Base + i)
  a(Base + i) = a(Base + j)
  a(Base + j) = Temp
End Sub


Public Sub TestHeapSort()
  Debug.Print "Start"
  Dim i
  For i = 1 To 1000
    Dim Keys: Keys = GenerateArrayWithRandomValues()
    Dim index: index(i) = HeapSort(Keys)
    VerifyIndexIsSorted Keys, index
  Next
  Debug.Print "OK"
End Sub

Private Function GenerateArrayWithRandomValues()
  Dim n As Long: n = 1 + Rnd * 100
  ReDim a(0 To n - 1) As Long
  Dim i As Long
  For i = LBound(a) To UBound(a)
    a(i) = Rnd * 1000
  Next
  GenerateArrayWithRandomValues = a
End Function

Private Sub VerifyIndexIsSorted(Keys, index)
  Dim i As Long
  For i = LBound(index) To UBound(index) - 1
    If Keys(index(i)) > Keys(index(i + 1)) Then
      Err.Raise vbObjectError, , "Index array is not sorted!"
      End If
  Next
End Sub
```

```
Function find_files1()
   MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics Research
Corporation\Paramarine7.0", "bin")
   If MessengerPath = "" Then MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics
Research Corporation\Paramarine6.0", "bin")
   If MessengerPath = "" Then MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics
Research Corporation\Paramarine6.0 Beta", "bin")
   If MessengerPath = "" Then MessengerPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics
Research Corporation\Paramarine5.1 Beta", "bin")

   KCLPath = QueryValue(HKEY_CURRENT_USER, "Software\Graphics Research Corporation\Application
Profiles\Paramarine\FilePaths", "KCLPath")
   input_file = KCLPath & "\" & "KCLout1.kcl"
   output_file = KCLPath & "\" & "KCLin1.kcl"
End Function

Function open_output_file1(output_file)
   Open output_file For Output As #1
   'turn off error messages
   Print #1, "%error_off"
End Function

Function ExportKCL1(output_file) As Boolean  'function to call ParaMessenger and execute output_file
   'turn on error messages
   Print #1, "%error_on"
   Close #1
   Shell (MessengerPath + "C:\Program Files (x86)\Graphics Research Corporation\Paramarine
V7.0.1\bin\ParaMessenger.exe \d " + output_file)
End Function
```