Section 3

PROGRAM OVERVIEW

3.1 INTRODUCTION

The Advanced Surface Ship Evaluation Tool (ASSET) is a family of interactive computer programs for use in the exploratory and feasibility design phases of surface ships. A distinct, but similar, program exists for each of several types of surface ships, including monohull surface combatants, monohull auxiliary and amphibious ships, monohull aircraft carriers, small waterplane area twin-hull (SWATH) ships, surface effect ships (SES), and hydrofoil ships. All of these programs are controlled by one common, Windows-based user executive program.

ASSET Family Of Programs

Monohull Surface Combatant Ships	MONOSC
Monohull Amphibious and Auxiliary Ships	MONOLA
Monohull Aircraft Carriers	MONOCV
Small Waterplane Twin Hull Ships	SWATH
Hydrofoil Ships	HYDROFOIL
Surface Effect Ships	SES

Within each ship-type-specific program there exists a series of computational modules. **These computational modules comprise the heart of every ASSET program.** Each module performs computational functions pertaining to a specific domain of naval architecture, such as hull geometry, hull structure, resistance, propulsion, machinery, weight, hydrostatics, seakeeping, etc. (Details of the various computational modules are addressed in Section 3.5.) These computational modules are powerful tools because their function is twofold. The bulk of the modules, categorized as synthesis modules, provide design synthesis capabilities. These modules form the core of the program's capabilities. Synthesis modules, when executed together, take user input data and, through an iterative process, alter design parameter values until the synthesis modules converge on a mathematically coherent and balanced design. Essentially, ASSET's synthesis capability performs a "design spiral." In addition to synthesis modules, other modules provide analysis of the current ship design (called "analysis modules"). Analysis modules take the ship design generated by the synthesis modules as input and return computer-calculated information pertaining to ship performance, seakeeping, stability, etc.

The following sections cover ASSET's conceptual structure and show how each of ASSET's elements combine to form a powerful, yet user-friendly ship design and evaluation tool. In addition, ASSET's user commands and computational modules are described. Understanding the structure of ASSET, the commands used to drive the program, and the modules used to perform the calculations is paramount in learning how to create sound, ASSET-aided, conceptual ship designs.

3.2 CONCEPTUAL ORGANIZATION

The ASSET ship design system is composed of eight principal elements:

- The user
- An executive program (the Windows-based user interface)
- A ship-type-specific ASSET design program (selected from the ASSET family of programs)
- A series of modules (for input/output, design synthesis, and analysis)
- A ship design undergoing generation or analysis (called "current model")
- A data bank (where the user's ship design data is stored)
- On-line documentation and help
- Hardcopy documentation

Figure 3.2 shows a pictorial layout of the ASSET system. Simply stated, the user, via an all-controlling executive program, generates and manipulates a current ship design by entering design data and running the selected program's modules. Each program within ASSET has its own computational modules specific to the program's ship-type capability. Ship designs and selected components of designs are stored in a storage facility called a data bank. During the design process, designers can define a ship using portions of pre-existing ship designs (contained in one or more data banks), manually input design data, have ASSET's modules calculate specific design parameters (where allowed), or any combination of the above.

Regardless of the user's program choice within ASSET's family of programs (monohull surface combatants, monohull auxiliary and amphibious ships, monohull aircraft carriers, SWATHs, SES, or hydrofoil ships), the same Windows-based user interface (executive program) is used.

Note: Although ASSET is a family of ship design programs, this manual, when discussing a specific program (such as MONOSC, MONOLA, etc.), will refer to the program as "ASSET." This is used for convenience purposes and is also meant to imply that many options and functions available in one program are available in the others.

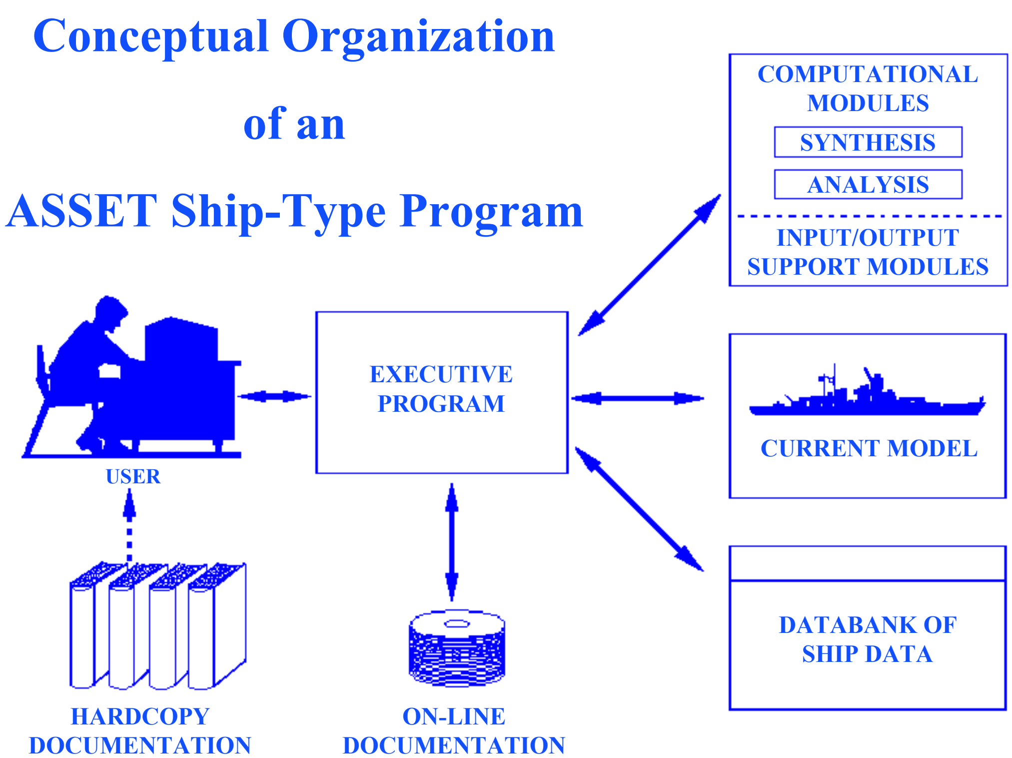


FIGURE 3.2 - ASSET CONCEPTUAL LAYOUT

3.2.1 User

The user is the controlling element of the ASSET ship design system. Through menu choices and/or direct program commands, the user directs the execution of, and interaction between, ASSET's components. **It is the user's responsibility to select proper input data and to effectively interpret ASSET's output.**

Like any computer program, the quality of ASSET's output is dependent on the user-supplied input. Ill-considered and erroneous input data will yield a product far from the original goal-- placing a critical burden on the user. The user needs to use discretion in selecting input data, making sure that the data (whether from a pre-existing ship in the data bank or from other sources) pertain to the ship under development and that the user's design requirements fall within the computational means of ASSET.

Keeping in mind the concerns mentioned above, a naval architect or any other technical individual familiar with ship design is needed to effectively create ASSET-aided, conceptual ship designs. Although ASSET is a powerful design tool, ASSET can not design a ship. The user is required to enter appropriate design data, accurately interpret ASSET output, and make well-informed design choices. Section 5 and 6, describe the recommended process to effectively employ ASSET as a ship design tool.

3.2.2 Executive Program

The executive program is the user's interface to all other aspects of ASSET. Using the executive program (user interface), the user can select a ship-type-specific program, direct data to and from data banks, run design synthesis and analysis modules, direct program output, etc. The executive program is Windows-based and is driven by commands found in drop-down menus. In addition to entering commands via menus, commands can be entered in the command text box located in the upper right corner of the main screen. ASSET commands are discussed in Section 3.4.

All of ASSET's capabilities can be accessed through the executive program. Additionally, regardless of the program chosen by the user, the same executive program is used. These two features help to make ASSET more user-friendly. This is accomplished by reducing the effort required to learn the use of more than one program from within ASSET's family of ship-type-specific programs.

3.2.3 Current Model

Simply stated, the current model is the ship design with which the user is working. Whenever a module is run, the required input data are extracted from the current model by the executive program and are used to generate output data. The current model can be altered in several ways. The user can add to or change the current model by manually entering data, importing data stored in a data bank, or by directing the program to generate model data. In the case of computer generated data, this usually means running one or more synthesis modules. Whenever an aspect of the current model is altered, the new data replaces the old data. The current model must be stored in a data bank to "save it to disk."

3.2.4 Data Bank

The data bank is where ship models (or "designs") are permanently stored. A data bank can store entire models or portions of these models, such as a propulsion plant or a hull. Storing a model or a portion of a model in the data bank is analogous to saving a file to disk. One can consider a data bank as a library in which ship models are stored. In addition to saving current models to the data bank, one or more data banks can be accessed during an ASSET session to import stored design data to the current model.

3.2.5 Modules

Each program within ASSET contains a selection of computational modules. **These computational modules comprise the heart of every ASSET program.** Each module performs computational functions pertaining to a specific domain of naval architecture, such as hull geometry, hull structure, resistance, propulsion, machinery, weight, hydrostatics, seakeeping, etc. These computational modules are powerful tools because their function is twofold. The bulk of the modules, categorized as synthesis modules, provide design synthesis capabilities. These modules form the core of the program's capabilities. Synthesis modules, when run together (the user is said to be “running synthesis”), take user input data and, through an iterative process, alter design parameter values until the synthesis modules converge on a mathematically coherent design. Essentially, ASSET’s synthesis capability performs a “design spiral.” In addition to synthesis modules, other modules provide analysis of the current ship design (called “analysis modules”). Analysis modules take the ship design generated by the synthesis modules as input and return computer-calculated information pertaining to ship performance, seakeeping, stability, etc. Analysis modules do not modify the design, and therefore do not output data to the current model.

It should be noted that synthesis modules could be run singularly as well as together. A primary purpose of running a synthesis module singularly is to facilitate the input of required data. When a synthesis module is run without all the data needed for computations, ASSET can follow three different means of obtaining this information. ASSET can supply temporary values, assign programmed default values, and/or query the user for the needed information. (See Section 3.4.2.3, *Command Strings*, for additional information.) When querying the user for data, ASSET will enter “prompt mode” and ask the user for data, item by item. This releases some burden from the user, freeing the user from determining and entering the hundreds of required datum entries before running the program. In other words, the user can wait to enter data until ASSET requires that particular information.

If the user is working with a current model which has been “mathematically converged” (i.e. it has already been run through synthesis) and the model is changed in any way (manually changing design parameters, etc.), the user must re-run synthesis to accommodate the newly entered information in the current model. Since design parameters are linked to other parameters, making a change in one parameter is consequential to a variety of other parameters (e.g. a change in payload will change ship weight and/or arrangement which in turn could change stability, powering requirements, etc.), “synthesis” must be re-run to adjust as many parameters as the new data affects.

In addition to computational modules, an export utility can be used to output data to non-ASSET programs in the required format of the non-ASSET program.

Section 3.5 contains a description of each module contained in all ASSET programs.

3.2.6 On-line Documentation and Help

Documentation and help functions are available to the user under the *Help* menu. The information available to the user includes definitions of all ASSET input parameters, descriptions of command functions and syntax, the ASSET System Manual, providing general information on ASSET capabilities and information of the menus, toolbars and other standard Windows capabilities.

3.2.7 Hardcopy Documentation

In addition to the on-line documentation and help, an extensive set of hardcopy documentation exists. Volumes pertaining to the theory behind ASSET's computational modules (analysis and synthesis) have been maintained. Additionally, documentation about ASSET's commands and model parameters exists.

3.3 DATA STRUCTURE

Central to the use of ASSET is an understanding of the data elements contained within a current model and the organization of those data elements. Each ship model contains hundreds of data that describe its characteristics. In order to navigate through ASSET, not only should the user know what type of information ASSET requires and/or generates, but also how ASSET organizes these data.

3.3.1 Model Parameter List

To simplify the data management task for ASSET, a powerful data management system has been installed. This data management system provides a mechanism for the transfer of data from the user to the program, from the program to the user, or internally within the program. The heart of the data management system is the model parameter list (MPL), which is a list of parameters (elements of data) used as input by one or more of ASSET's computational modules (synthesis and analysis). Each parameter (element of data) is assigned a name composed of one or more words and abbreviations. This parameter name is used to assign a value(s) to the given element of data, to query the data element for its value, to prompt (or query) the user for specific data input, and for other data management tasks.

Organization of the current model's data is accomplished by a mutli-tier, tree-type hierarchy, known as the model parameter list (MPL). The highest tier of the MPL represents the entire ship, or current model. The next tier consists of primary groups, which may further be subdivided into secondary groups, which may be subdivided into narrower tertiary groups, etc. Finally, the lowest-level groups are subdivided into parameters, which are the lowest level of the hierarchy. **The only level in the hierarchy to contain actual data is the parameter level; all other tiers are merely names by which a tree-like organization is formed.** These tier names are most general on top and become specific as one descends each tier, with the most specific being the actual parameters that receive data. The organization of the ASSET groups is intended to loosely follow the Ship Work Breakdown Structure (SWBS).

Figure 3.3 shows an abbreviated model parameter list hierarchy utilized by ASSET. The system level represents the entire ship. The groups represent major physical elements of the ship, such as hull, superstructure, propellers, and major areas of analytical concern, such as mission requirements, and cost factors.

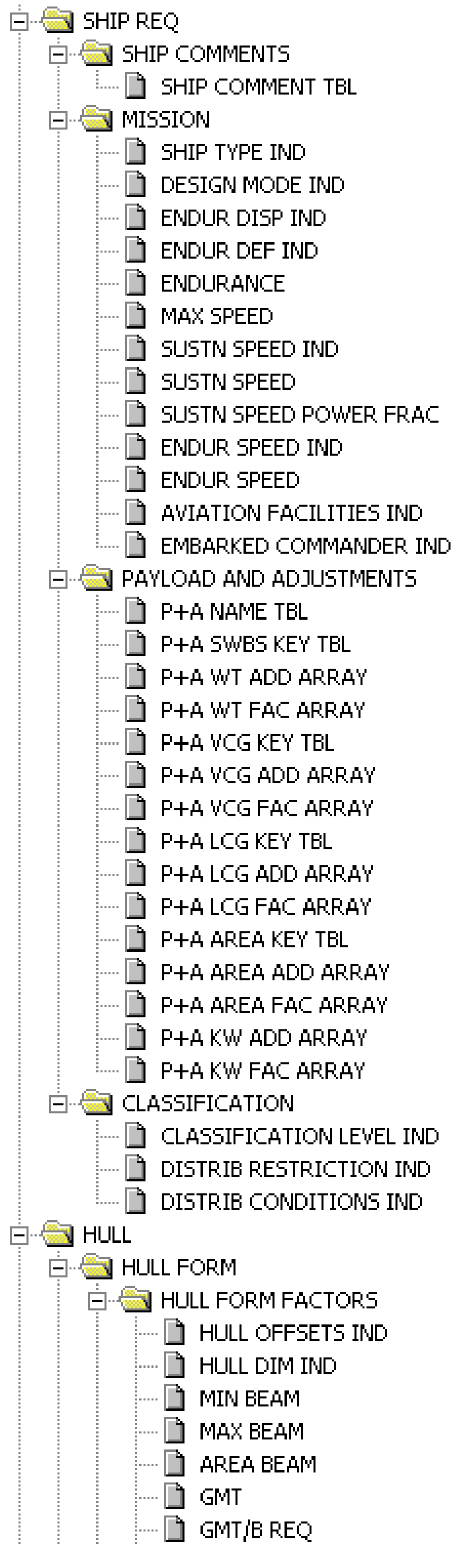


Figure 3.3 - MPL hierarchy example

In Figure 3.3 two primary groups are shown, the SHIP REQ group and HULL group. Secondary groups to the SHIP REQ primary group include SHIP COMMENTS, MISSION and PAYLOAD AND ADJUSTMENTS. The HULL primary group contains a secondary group HULL FORM that contains a tertiary group HULL FORM FACTORS. Finally, the actual data reside in the parameter tier of the MPL hierarchy. Parameters within the HULL FORM FACTORS tertiary group are HULL OFFSETS IND, HULL DIM IND, etc. As shown in Figure 3.3, each of these parameters has a specific value associated with it. A description of each group and their parameters can be found in the on-line help.

3.3.2 Data Types

Parameters may represent any of four data types: a single number (scalar), a matrix of numbers (array), a single character string (indicator), or a matrix of character strings (table). A specific type of parameter exists for each data type, and are called scalar parameters (single number), array parameters (matrix of numbers), indicator parameters (single character string), and table parameters (matrix of character strings). An example of a scalar parameter is the main engine specific fuel consumption (MAIN ENG SFC), which equals the specific fuel consumption of the main engines at their maximum continuous power. This parameter has a single value. An example of an array parameter is the main engine dimension array (MAIN ENG DIM ARRAY). This array parameter contains the length, width, and height of the main engine. This single parameter therefore represents three different numbers in the form of a matrix. An example of an indicator parameter is the main engine type indicator (MAIN ENG TYPE IND). Indicator parameters are typically used to instruct a computational module to do processing corresponding to one of several options, thus the indicators control the flow through the program. For example, the main engine type indicator may be assigned any one of the following values: GT, RGT, D DIESEL, F DIESEL, 600 ST, 1200 ST, or NUCLEAR. These values correspond to gas turbine, regenerative-cycle gas turbine, domestic diesel, foreign diesel, 600 psi steam turbine, 1200 psi steam turbine, or nuclear engine types, respectively. This parameter is used to order the computational modules to use algorithms for one engine type instead of those for any of the other types. An example of a table-type parameter is the payload and adjustment name table (P+A NAME TBL). This parameter contains the names of each of the many military payload and adjustment items (such as guns, helicopters, missiles, etc.) carried onboard ship. The names are contained within one or more strings of characters, arranged in a matrix.

3.4 EXECUTIVE COMMANDS

Executive commands are the instructions given by the user to perform various program functions. Since the executive program is Windows-based, all commands can be issued by choosing them from a series of drop-down menus. In addition to the menus, commands can be given by typing them in the command text box. When typing ASSET commands, one must follow the format recognized by ASSET. Following is a discussion on ASSET's commands and means of input.

3.4.1 Form

If the user employs the command text box to issue commands, the user must follow the format recognized by ASSET. This format is called a command string. Each command string is composed of one or more phrases, with each phrase within a command string separated by a delimiter, such as a comma or equal sign. To speed entry of commands, each phrase may be abbreviated. ASSET reads one command string at a time, processing each phrase within the command string.

An example of this command logic, but not a valid command string, is: BUY, COOKIES, 1 DOZEN. In this string, the command is to buy one dozen cookies. The first word/phrase defines the action-- buy. The second word/phrase answers the question, buy what? And the third word/phrase answers the most logical next question, how many? The first phrase of a command string is always a verb, such as SET, SHOW, or RUN. In the above example, BUY would be the verb (although not a recognized ASSET command). Presently, there are eighteen different verbs with which to begin a command string (see section 3.4.3). Most verbs require that one or more phrases follow the verb phrase. The phrase immediately following the verb phrase is an object phrase. In the above example, the object phrase is COOKIES. The object phrase defines the direct object of the verb's "action." Additional phrases may follow the object phrase depending on the nature of the first two phrases. These additional phrases are called modifier phrases, as they usually clarify the object. In the above example, 1 DOZEN clarifies the command to buy cookies.

Shown below is an example of a valid ASSET command.

SET, LBP, 500

For this example, the verb phrase is "SET", the object phrase is "LBP", and the modifier phrase is "500." This command string would cause the value within the current model of the hull length between perpendiculars to be given a value of 500 (feet).

Note: When using the Windows drop-down menus, the issuance of commands does not take the command string form. The benefit of these menus is that commands are simplified to a "point and click" and may seem easier to operate than issuing commands in the command text box. After choosing a menu command, the command-string-equivalent is generated in the command text box. The user interface takes the menu commands and translates them into valid command strings. After operating ASSET for some time, one might find it easier to issue some commands in the form of command strings in the command text box.

3.4.2 Input Modes

There are several means to send commands and enter data. The user can issue commands through the pull-down menus, enter data via ASSET's editor, or issue commands and input data through command strings.

3.4.2.1 Menu Choices

The primary form of input is the drop-down menu choices from the executive program's Windows-menus. These menus can perform many of the ASSET functions and create a user-friendly work environment. A tour of ASSET's menus is located in Section 4.3. Windows menu choices can be used to enter parameter values by selecting a particular parameter or group of parameters and then entering the ASSET editor for direct input.

3.4.2.2 ASSET Editor

The ASSET editor is the most efficient way to enter large arrays or large quantities of parameter values. The ASSET editor follows a spreadsheet style that contains all the parameter values of the current model. One can add or change parameter values by entering the spreadsheet editor and typing values via the keyboard. The power of the editor is that parameters can be set without the issuance of a command string (such as SET, LBP, 500). In the editor, the user can scroll or jump to the subject parameter and directly type its value.

3.4.2.3 Command Strings

Command strings are groups of phrases linked together to issue commands to the executive program. For example, in the previous example the parameter LBP was given a value by issuing the command, SET. In addition to setting parameter values, command strings can be used for a variety of other purposes, such as running a module, directing output, etc. Command strings are issued through the command text box.

When issuing a command string, two methods may be used. The first method, which has been discussed previously, is to explicitly type the full command (SET,LBP,500). The second method is called "query" or "prompt" mode. The user typing only a portion of a full command and then substituting the remaining portion with a question mark (?) initiates this method. The question mark tells ASSET to enter query mode and subsequently ASSET will display a menu of remaining choices for the partial command given. For example, if the user wants to set the type of main engine, but is unsure of the accepted input, the user can type: SET, MAIN ENG TYPE IND,?. This input will yield a menu displaying the remaining available options from which to choose. Each option has a menu number and can be selected by typing the corresponding number or actual value at the prompt.

Another means of entering prompt mode is available. When the user runs a module without supplying all the needed parameter values, ASSET will do any combination of the following three actions. One, ASSET will assign temporary values to the empty parameters; two, ASSET will assign default values to the empty parameters; and three, ASSET will enter prompt mode and query the user for the needed data. Figure 3.4 shows the results of running a module with missing data.

```
  RUN, HULL SUBDIV MODULE
** FATAL ERROR - HULL SUBDIV MODULE ** (E-INVALIDDATA-HSDMPL)
THE FOLLOWING PARAMETERS CONTAIN INVALID OR MISSING DATA:
    TRANS BHD SPACING                  SHAFT DIA ARRAY
    MECH ARR NO ARRAY

** ENTERING PROMPT MODE **
ENTER 'QUIT' TO RETURN TO COMMAND LEVEL.
```

Figure 3.4 Entering Prompt Mode to Supply Missing Data

Note: Temporary values are data set by ASSET to allow the specified module to run. When the module is completed, these values do not become part of the current model- the corresponding parameters in the current model are still at their "no data" state.

Temporary values are supplied because the specified module is not the appropriate module to assign permanent values to the parameters in question. Default values are data set by ASSET to allow the specified module to run, but are different than temporary values- default values become part of the current model. Default values are data coded in the program that represent commonly used values. The user may change these default values at any time. When a module is completed, ASSET warns the user of any temporary and/or default values that were assigned. See Figure 3.5.

```
  RUN, HULL SUBDIV MODULE
** WARNING - HULL SUBDIV MODULE ** (W-DEFAULTVALUES-HSDMPL)
THE FOLLOWING PARAMETERS WERE PROVIDED DEFAULT VALUES:
    SHIP TANKAGE KG ADD             SHIP TANKAGE KG FAC
    MR BHD SEP ARRAY                MR AFT BHD LOC
    SHAFT ANGLE ARRAY
** WARNING - HULL SUBDIV MODULE ** (W-TEMPVALUES-HSDMPL)
THE FOLLOWING PARAMETERS WERE PROVIDED TEMPORARY VALUES:
    MR LGTH REQ ARRAY               MR HT REQ ARRAY
    TANKAGE VOL REQ
```

Figure 3.5 Temporary and Default Value Warnings

3.4.3 Command Summary

Currently, there are seventeen verbs that are recognized by the Executive program. They are: ATTACH, COMMENT, EDIT, EXIT, INCLUDE, MODIFY, READ, REINITIALIZE, REMOVE, REWIND, RUN, SET, SHOW, SKIP, STORE, USE, and WRITE. Through the use of the Windows-menus, many of these commands never need to be typed. Choices from the pull-down menus correspond to these commands. For example, if one wanted to load a pre-existing ship named FRIGATE from the current data bank, one would click on the FILE menu and then choose OPEN. A standard Windows dialog box would appear and the name FRIGATE would be selected from the list of available ships (stored in the data bank). These menu options take the place of issuing the command string USE, FRIGATE, ALL.

A description of all seventeen commands follows. See Section 4.3, *Tour of the Menus*, for a Windows-menu equivalent of these commands. The commands listed below are grouped according to function. Each command performs a particular function in the four following categories: program control, data bank, current model, and session log/data transfer.

Program Control

The EXIT command is the command used to terminate the execution of the program.

The RUN command causes execution of a module, of the entire synthesis process, or of some auxiliary program

The SKIP command is used to remove a synthesis module from active status as a part of the synthesis module loop. Any synthesis module that has been removed from active status will not be run during the execution of the synthesis process. See comment under the INCLUDE command.

The INCLUDE command is used to restore a synthesis module to active status, should it have been deactivated previously during a given ASSET session. Only those synthesis modules whose status is active are executed during the synthesis process. **This is a good example of the benefit of the Windows-menus.** Using a Windows-menu, at a glance, one can see which modules are active and toggle any module to the off/on position.

Data Bank

The ATTACH command allows the user to switch from one data bank to another during an ASSET session. When a different data bank is selected with the ATTACH command, the data in the current model is not changed. At the start of an ASSET session, the user needs to attach a pre-existing data bank or create a new one. ASSET prompts the user for this input.

The USE command transfers specified data from the data bank to the current model. Previously existing data within the current model are overwritten with the corresponding new data from the data bank.

The STORE command saves data as a ship or component to the data bank. The data is retained (saved) in the data bank until it is removed by explicit request of the user. The STORE command is used to save a ship or component under a new name in the data bank. See MODIFY.

The MODIFY command replaces specified data bank data with current model data- in effect, saving the current model, or a portion of it, to a ship or component name already contained in the data bank.

The REMOVE command deletes a ship or component from the data bank.

Current Model

The SET command allows the user to input values to the current model, or to input values for program flags or switches.

The SHOW command allows the user to output various values. Data bank data, current model data, and the values of program flags and switches are among the values that can be output.

The EDIT command invokes the ASSET editor, which allows the current model's parameter values to be edited.

The REINITIALIZE command causes the program to reset the current model, a group of parameters, or an individual parameter to the no-data value, to reset all program flags and switches to their initial state, or to clear the contents of the log file.

Session Log and Data Transfer

The READ command causes command strings to be input from a specified off-line file. After the entire file has been read, the input of commands from the user is resumed.

The WRITE command causes specified data bank data to be written to a specified file in input-recognizable format. This command is typically used in conjunction with the READ and REWIND commands to transfer data from one data bank to another.

The COMMENT command allows the user to enter a string of text that will be treated by the program as commentary. Such commentary may clarify a user's thought process when the ASSET session is reviewed at a later date.

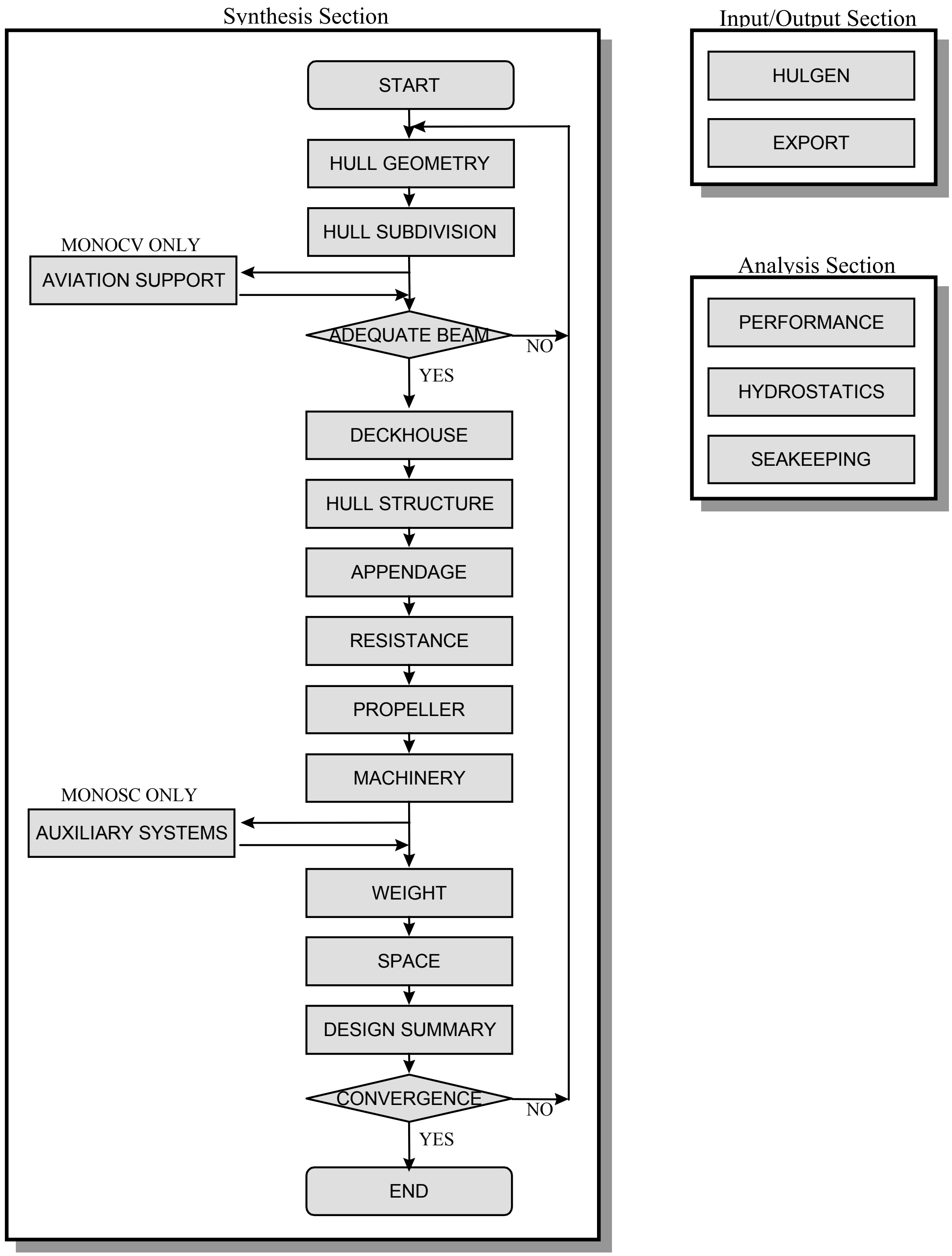
The REWIND command rewinds a specified file.

3.5 PROGRAM MODULES

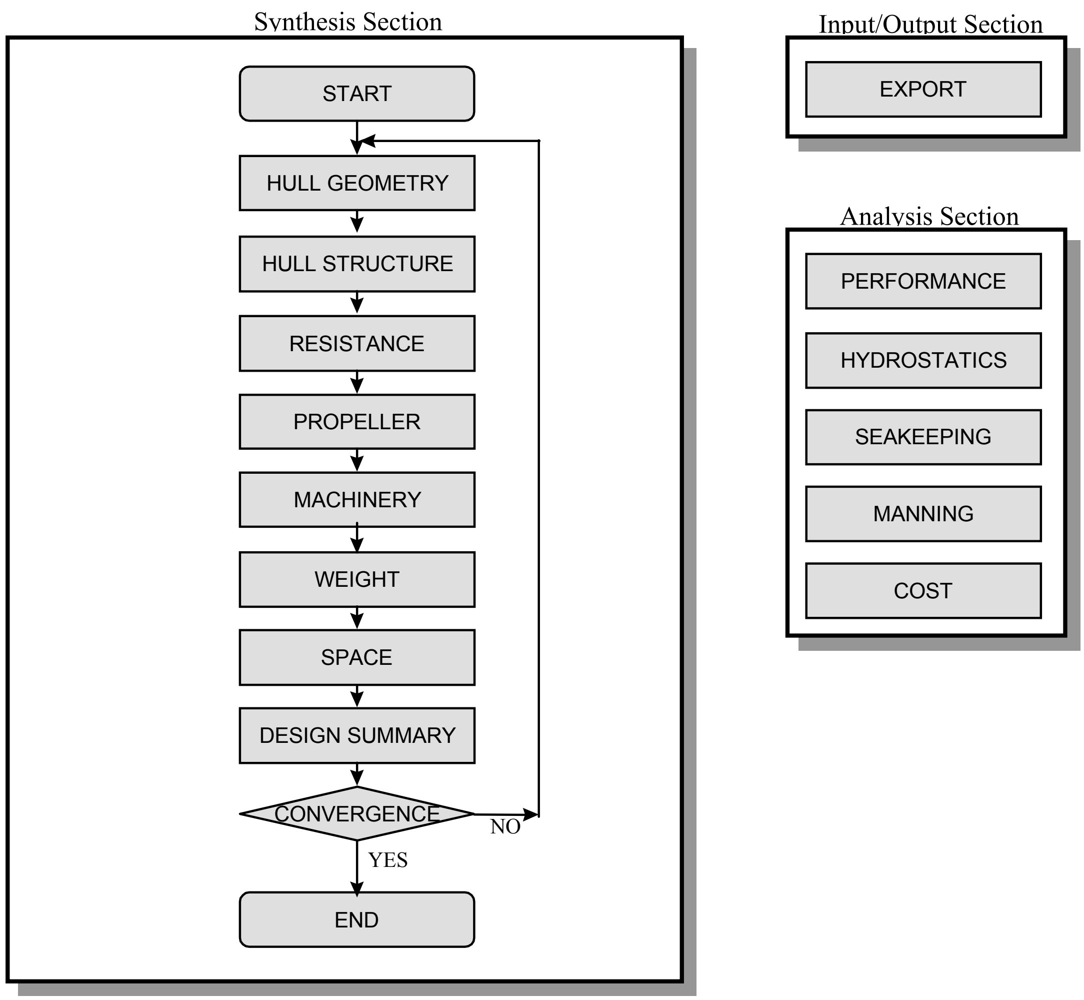
ASSET's computational modules comprise the heart of its capabilities. The computational modules are divided into synthesis and analysis modules. As described in Section 3.2.5, the synthesis modules take user input data and, through an iterative loop, alter existing parameters until the synthesis modules have converged on a coherent design. A model has converged when two runs through the synthesis loop yield identical designs (i.e. within the pre-set synthesis tolerance). The analysis modules are used after synthesis has been successfully performed. These modules calculate numerous design characteristics using the current model parameters. The synthesis and analysis modules, as well as auxiliary input/output modules are described in the following sub-sections.

The following figures show the module flow charts for each program within ASSET's family of programs. Notice the synthesis section and how the synthesis modules are linked together to form the characteristic iterative loop. After the model has converged, the model can undergo analysis by any or all of the analysis modules. Many modules are common to all of ASSET's programs, but some vary as dictated by the theory and nature of the type of subject vessel.

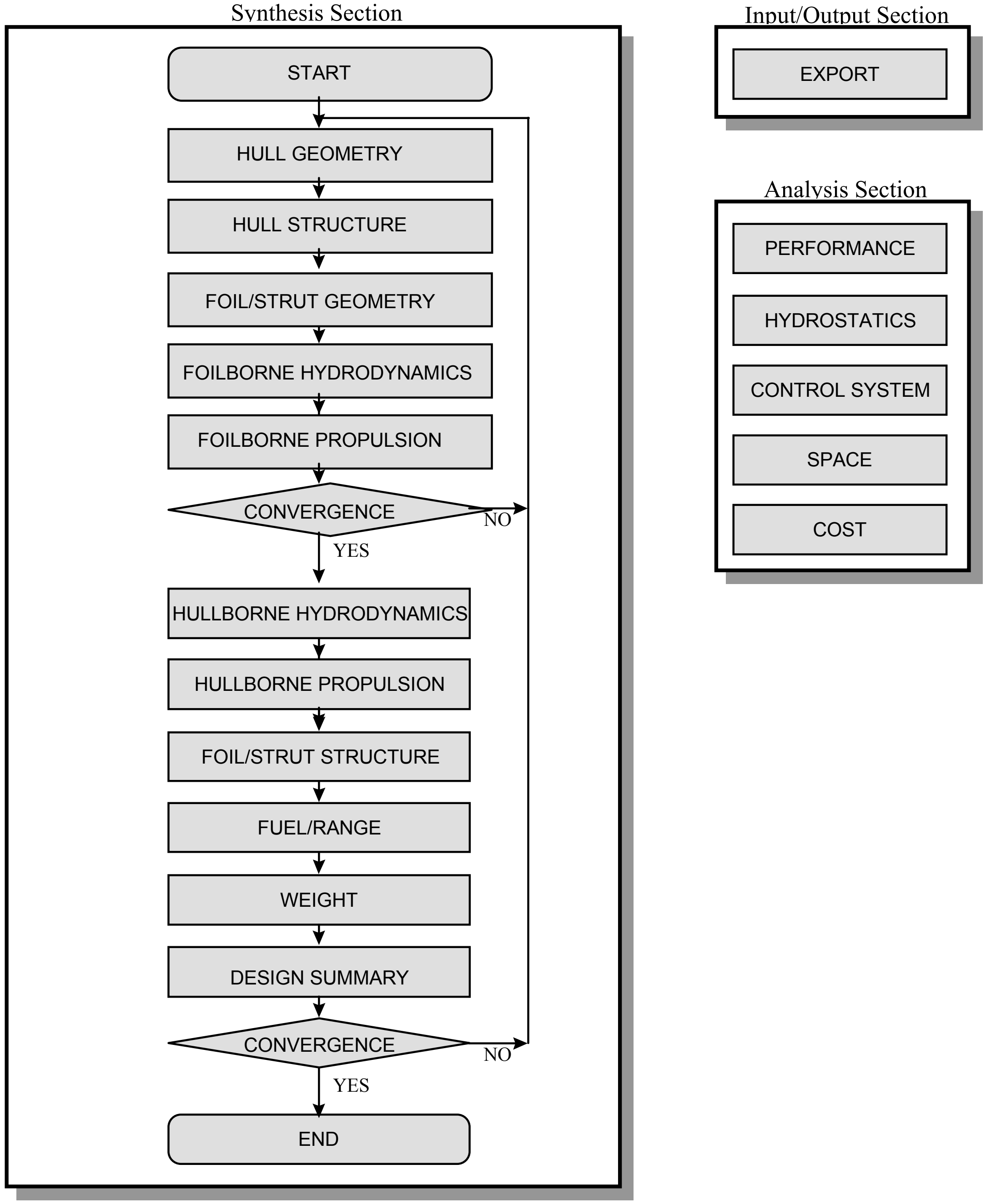
MONOSC/MONOLA/MONOCV Modules



SWATH Modules



HYDROFOIL Modules



3.5.1 Synthesis

Following is a descriptive list of the synthesis modules found within ASSET's family of programs.

Hull Geometry Module

The Hull Geometry Module defines the geometry of the molded hull form via a set of offsets. Offsets may be: 1) input to the module and used "as is;" 2) input and then scaled and warped to define a new hull form that meets requested physical characteristics; 3) generated by the module as a function of a user-specified parent hull form. This module is located in all ASSET programs.

Hull Subdivision Module

The Hull Subdivision Module defines the internal hull subdivision, including decks, platforms, transverse bulkheads, longitudinal bulkheads, and inner bottom. The position of decks, platforms, and transverse bulkheads may be given by the user, or the module may be directed to automatically position these elements. This module also calculates the available machinery room volume, tankage volume, and internal arrangeable deck area. This module is located in MONOSC, MONOLA, SWATH, and MONOCV programs.

Deckhouse Module

The Deckhouse Module defines the geometry of the ship superstructure via the use of prismoids. Prismoid geometry may be: 1) input to the module and used "as is;" 2) revised to form a new superstructure of requested characteristics; 3) automatically generated as a function of various requirements. This module also calculates the internal deck area of the deckhouse and the deckhouse weight. This module is located in MONOSC, MONOLA, SWATH, and MONOCV programs.

Hull Structure Module

This module calculates scantling and weight data for the ship elements defined in the current model. The calculations are based on design bending moments and pressure loading characteristics that are either calculated by the program or input by the user. Scantlings are determined at the midship section for the hull bottom, hull sides, and weather deck. Additional scantling data are calculated for the inner bottom, internal decks, bulkheads, frames, girders, beams, and stiffeners. This module allows a variety of structural arrangements, and also determines the scantlings for the least weight, if requested. This module is based on the Navy's *Structural Synthesis Design Program* (SSDP). This module is located in all ASSET programs except Hydrofoil and SWATH. The Hydrofoil and SWATH programs have unique hull structure modules.

Appendage Module

The Appendage module defines the geometry of hull appendages (except propulsion-related appendages) and computes their displacements. Appendages treated by the module include the skeg, bilge keels, sonar dome, shell plating, and roll fins. This module is located in MONOSC, MONOLA, and MONOCV programs.

Resistance Module

The Resistance module calculates ship resistance. Calm seas and a clean hull are assumed. The total ship resistance is computed as the sum of frictional, residuary, appendage, and wind resistance along with a resistance margin. Taylor series data as modified by the application of a worm curve, a destroyer regression analysis, NRC series data, or user defined residuary resistance coefficients are used to calculate residuary resistance. Frictional resistance is computed using either the ATTC or ITTC friction line. This module is located in all programs except for the Hydrofoil and SWATH programs. The Hydrofoil and SWATH programs have unique resistance modules.

Propeller Module

The Propeller module determines the geometric characteristics of the ship propeller and calculates the power that must be supplied to the ship's propeller(s) to achieve various ship speeds. The user may select a propeller data series or the module can be directed to use propeller data that are input to the module. Propeller data that are internal to the module include the Wageningen B-screw (Troost) series and two regressions developed by the Navy utilizing lifting-line computations. This module is located in all programs except for the Hydrofoil program. The Hydrofoil program has unique propulsion modules.

Machinery Module

The Machinery module computes electric power requirements and performs sizing and arrangement of the following ship elements: main propulsion machinery, secondary propulsion machinery, and electric plant. This module also establishes the characteristics of the transmission, whether mechanical or electrical. The amount of fuel that must be carried onboard ship to achieve a specified endurance or the achievable endurance for a given amount of fuel is also calculated. This module is located in all programs except for the Hydrofoil and SWATH programs. The Hydrofoil and SWATH programs have unique machinery modules.

Weight Module

The Weight Module calculates a detailed weight breakdown for the ship. The weight statement follows the Navy Ship Work Breakdown Structure (SWBS). A weight module is located in all ASSET programs; however, the algorithms used within the module vary with each ship type.

Space Module

The Space Module calculates the total area and volume requirements of the ship. The space statement follows the Navy Ship Space Classification System (SSCS). This module is located in all programs except for the Hydrofoil Program.

Foil/Strut Geometry Module

The Foil/Strut Geometry Module sizes foils, struts, and pods in accordance with the defined hull size, and with the foil system type and geometric data provided. Single T, double T, PI, or three-strut configurations may be used for the aft and forward foil systems. Longitudinal locations of struts are calculated from a foil loading ratio specified by the user. This module is located only in the Hydrofoil Program.

Foilborne Hydrodynamic Module

This module uses hull and foil system data to calculate foilborne drag and takeoff drag. This module is located only in the Hydrofoil Program.

Foilborne Propulsion Module

This propulsion module performs sizing calculations for either a foilborne-waterjet or foilborne-propeller propulsion system. The waterjet-propulsion system section of this module calculates engine power requirements, water-duct losses, pump size, and operating data based on given drag, duct, and pump type data. The propeller-propulsion system section calculates engine power requirements, z-drive transmission parameters, propeller size, and propeller operating data based upon given drag, gearbox, and propeller characteristic data. This module is located only in the Hydrofoil Program.

Hullborne Hydrodynamic Module

The Hullborne Hydrodynamic Module calculates ship resistance data during hullborne operation. Either planing hull or Taylor Standard Series resistance-type calculations may be performed. This module is located only in the Hydrofoil Program.

Hullborne Propulsion Module

The Hullborne Propulsion Module calculations parallel those of the foilborne propulsion module except that all data for the hullborne propulsion system are sized. This module is located only in the Hydrofoil Program.

Foil/Strut Structure Module

The Foil/Strut Structure Module calculates scantlings of the primary load-carrying structure of the foils and struts. The calculations are based upon geometric data and loading conditions derived from hydrodynamic and inertial forces developed during foilborne operation. Loads include asymmetrical foil loading hydrodynamic lift distribution, and incremental lift, drag, and side loads associated with maneuvers and operation in a sea state. This module is located only in the Hydrofoil Program.

Fuel/Range Module

Range performance is calculated by this module in either two ways. The weight of fuel required to achieve a specified foilborne range is calculated, or the range that may be achieved by a given ship is calculated. The calculation mode is specified by the user. Fuel requirements for auxiliary and electric plants are also considered. This module is located in only the Hydrofoil Program.

Design Summary Module

The sole function of the Design Summary Module is to produce output to the user that summarizes the results of computations of the other synthesis modules. Output from the Design Summary Module is often more convenient to scan than output from each of the other synthesis modules. This module is located in all ASSET programs.

3.5.2 Analysis

Analysis modules provide design analysis of the synthesis-generated model in various aspects of Naval Architecture. Unlike synthesis modules, analysis modules are always run individually and, most importantly, do not alter the current model. Analysis modules take the current model as input to generate design characteristics such as performance, hydrostatics, seakeeping, manning, cost, etc. Following is a descriptive list of the analysis modules found within ASSET's family of programs.

Performance Analysis Module

The Performance Analysis Module calculates the performance characteristics of ship designs that have been generated via the synthesis process. Whereas synthesis performance calculations assume calm water and a clean ship, the Performance Analysis Module considers fouling effects of marine organism, degradation of machinery with time, and sea state operation. This module is located in MONOSC, MONLA, MONOCV, and SWATH.

Hydrostatic Analysis Module

The Hydrostatic Analysis Module determines the hydrostatic properties of the ship design. Data are calculated for hydrostatic properties of form, floodable length, intact stability, damaged stability, and maximum VCG positions allowed by NAVSEC Design Data Sheet DDS 079-1 criteria. This module includes portions of the Navy *Ship Hull Characteristics Program* (SHCP). This module is located in MONOSC, MONLA, and MONOCV.

Seakeeping Analysis Module

The Seakeeping Analysis Module calculates a relative ranking (on a scale of 10) of the seakeeping characteristics of the ship. The ranking is based on pitch and heave motions only. This module is derived from the work of N. K. Bales and W. R. McCreight at DTNSRC. The Seakeeping Analysis Module is only in MONOSC.

Control System Analysis Module

This module allows the user to obtain quantitative information regarding the dynamic stability and controllability of the foilborne ship in a sea state. A set of stability boundaries based upon foil system geometry is calculated to determine whether the boundaries are violated when the ship is exposed to several sea conditions. This module is located only in the Hydrofoil Program.

Space Analysis Module

The Space Analysis Module generates an estimate of the area and volume requirements of the ship. The space statement follows the Navy Ships Space Classification System (SSCS). This module is located only in the Hydrofoil Program.

3.5.3 Input/Output

The following capabilities are available to assist in the input of data and in the transfer of data to other design and analysis programs.

HULGEN Module

The HULGEN Module allows the user more control over the definition of the molded hull form. When the Hull Geometry Module is to generate a new hull form a set of control curves called boundary conditions are used to define the shape characteristics of the hull form. Several sets of boundary conditions are available to the Hull Geometry Module that emulates the shape of typical common hull forms. The HULGEN Module provides the user the capability to edit these boundary conditions to generate a unique hull form and to see the impact of each change as they are input. The HULGEN Module is available in the MONOSC, MONOLA, and MONOCV programs.

Export Utility

The Export Utility transforms ASSET data into several file formats that may be used as input to other CAD, design, and analysis programs. The Export Utility is not a single module, but is a utility which may call one or several of the ASSET modules to generate the data required for the particularly file format desired. The Export Utility is available in the MONOSC, MONOLA, and MONOCV programs.