Application of PC-Based Project Management in an Integrated Planning Process

U. S. DEPARTMENT OF THE NAVY
CARDEROCK DIVISION OF
THE NAVAL SURFACE WARFARE CENTER

in cooperation with
Newport News Shipbuilding
**Report Documentation Page**

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204. Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAY 1992</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application of PC-Based Project Management in an Integrated Planning Process</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naval Surface Warfare Center CD Code 2230-Design Integration Tools Bldg 192, Room 128 9500 MacArthur Blvd, Bethesda, MD 20817-5700</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release, distribution unlimited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT</td>
</tr>
<tr>
<td>b. ABSTRACT</td>
</tr>
<tr>
<td>c. THIS PAGE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. NUMBER OF PAGES</th>
<th>199</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
</table>

Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std Z39-18
This report / manual was prepared as an account of U.S government sponsored work. Neither the United States, nor the Carderock Division of The Naval Surface Warfare Center - Carderock Div. NSWC (formerly The David Taylor Research Center-DTRC), nor any person acting on behalf of the Carderock Div. NSWC (a) makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness or usefulness of the information contained in this report / manual, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or (b) assumes any liabilities with respect to the use of or for damages resulting from the use of any information, apparatus, method, or process disclosed in the report / manual. As used in the above, “persons acting on behalf of the Carderock Div. NSWC” includes any employee, contractor, or subcontractor to the contractor of the Carderock Div. NSWC to the extent that such employee, contractor or subcontractor to the contractor prepares, handles, or distributes, or provides access to any information pursuant to his employment or contract or subcontract to the contractor with the Carderock Div. NSWC. ANY POSSIBLE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR PURPOSE ARE SPECIFICALLY DISCLAIMED.
APPLICATION OF PC-BASED PROJECT MANAGEMENT IN AN INTEGRATED PLANNING PROCESS

Prepared by RICHARD J. NEUMANN DAVID J. McQUAIDE

For NEWPORT NEWS SHIPBUILDING AND DRYDOCK CO.

In behalf of SNAME SHIP PRODUCTION COMMITTEE PANEL SP-8 on INDUSTRIAL ENGINEERING

Under the NATIONAL SHIPBUILDING RESEARCH PROGRAM

May 1992
ACKNOWLEDGEMENTS

This is the final report of a project developed and cost shared by National Steel and Shipbuilding Co. (NASSCO) under a David Taylor Research Center contract. The program is a cooperative effort of the U.S. Navy and the United States shipbuilding industry. Industry direction was provided by the Society of Naval Architects and Marine Engineers’ (SNAME) Ship Production Committee on Industrial Engineering (SP-8). The program manager is W.G. Becker of Newport News Shipbuilding.

The authors would like to thank Jim Royle, Jon Gribskov, Pete Jaquith, Len Schneider and John Lyle of NASSCO for believing in our ideas and allowing us the time to develop the system. We would also like to thank Lyn Haumschilt, Bill Wilson, and many others at NASSCO for their assistance, insight, and advice without which this paper would not have been possible. Appendix VI of this submittal is based upon the insights gained by a NASSCO team headed by Steve Eckberg. Steve and his team deserve credit for the excellent analysis of the applicability of stochastic simulation vs. deterministic project management software in schedule development. We are indebted to Bill Caughlin and Welcom Software for their generous training, support, and recommendations. We would also like to thank the SP-8 panel for taking the time to review and comment on this submittal. Your inputs have made this a better paper.
The report and software included within this package were developed under the auspices of SNAME Ship Production Committee panel SP-8 Task N8-91-6, "Application of PC-Based Project Management in an Integrated Planning Process". The task was performed by Richard J. Neumann and David J. McQuaide of National Steel and Shipbuilding Co., San Diego. This submittal consists of four parts as shown below:

- The "Project Report" which describes the system's philosophy and overview.
- The system's "Users Manual" which explains the operation of the Integrated Production Planning System as developed in this package. The manual assumes the user has a moderate knowledge of dBase programming along with an understanding of project management techniques and ship production planning.
- An appendix containing program logic flow charts, program coding and documentation, and disk copies of all programs and data files developed by the project for use by the system.
- A "System Demonstration Disk" which is an on-screen slide show presentation where the user can step through the system and see the various screens of the Integrated Production Planning System along with on-screen explanations.
EXECUTIVE SUMMARY

This task has developed a PC-based system which serves as a tool to assist planning organizations in developing, updating, and revising ship production schedules. The system will also create and update manning, facility, and material utilization reports. The scope of the system developed is limited to the ground assembly, outfit, join, and erect operations. The "Project Report" describes the data required by the system to produce its outputs. The report explains the system development philosophy and gives an overview of the schedule generation system. To demonstrate the use of the system, data for a sample ship is given and a schedule developed based upon this data.

The User's Manual serves as a reference for shipyards wishing to develop a PC-based Integrated Production Planning System (IPPS). The software included in this package is not intended to be a turnkey system. For an IPPS to work for a particular shipyard, the shipyard must modify the coding so that the system will conform to the yard's facilities and methods of operation. The IPPS should not be viewed as a computer system; it is a production planning system that makes use of computer tools. Simply obtaining and installing the software will not give a shipyard an operable system. Developing an Integrated Production Planning System is a significant task. However, once developed, the IPPS is a valuable tool that will assist shipyard personnel in making effective production decisions.
PROJECT REPORT

INTEGRATED PRODUCTION PLANNING SYSTEM
ABSTRACT

When a schedule is developed for a project it dictates not only the dates when various activities should occur, but also the manning, facility, and material utilization required to meet the schedule. A change in schedule necessitates a change in manning, facility and material utilization plans. By the same token, changes in manning, facilities or material availability (e.g. late arrival of material) necessitates a change in schedule. Since the activities within a project are interrelated and various projects often use common manning and facilities, a change in a single activity often necessitates the modification of many related activities.

A shipyard working on several projects must schedule thousands of interrelated activities. To remain credible, the schedules must be updated as items are delayed and variations in production schedules occur. For a shipyard to remain competitive, it must have a production planning system that optimizes the yard’s overall use of facilities and manning.

This paper discusses the development and implementation of a PC-based Integrated Production Planning System (IPPS) which serves as a tool to assist planning organizations in developing, updating, and revising schedules and associated manning, facility, and material utilization reports.
GLOSSARY

The following definitions are provided to clarify usage within this paper. They are not meant to imply any type of industry standard.

MASTER PRODUCTION SCHEDULE (MPS) - (as defined by the American Production and Inventory Control Society Dictionary) "...the anticipated build schedule for those selected items assigned to the master scheduler. . ." [1]

BLOCK - A structural assembly which will be outfitted and erected singly or as part of a grand block.

GRAND BLOCK - Two or more blocks that have been joined into a single interim product prior to erection.

LONG RANGE FORECAST - Needs implied by schedules over a two to three year time span. This forecast will show manning and capacity needs of a project for its entire ground assembly, join, outfit, and erect period.

SHORT TERM REQUIREMENTS - Needs implied by schedules over a two to three month time span. Requirements are used for regularly updated, detailed manpower and facility utilization planning.
SCOPE OF PROJECT

An effective integrated production schedule will consider all activities that go on within a shipyard. However, this does not mean that a single production planning system must model all activities. If an individual system models a well-defined area of the shipyard, this information can be combined with information regarding other areas to develop an overall view of the shipyard system.

At the highest level, a shipyard may be described as consisting of four interrelated functions: production, materials, engineering, and personnel/business administration/business development. Production operations may be grouped in a variety of ways. The grouping used in this paper is as shown in figure 1.

The scope of activities to be modeled by the IPPS discussed in this paper are limited to ground assembly, joining, outfitting, and erection of blocks. The activities to be modeled are as listed.
- Fabrication of Steel Parts
- Block Sub-Assemblies (i.e., building of bulkheads, decks, etc., from fabricated parts)
- Block Assembly
- Pre-Blast Outfitting of Blocks/Grand Blocks
- Blast and Paint of Blocks/Grand Blocks
- Post-Blast Outfitting of Blocks/Grand Blocks
- Grand Blocking (joining of blocks before they erect to ship)
- Block or Grand Block Erection to Ship

The on-board outfitting, shop, production service, repair, and non-production activities are not modeled by this system. Schedules and information regarding these activities are developed in parallel with this system. The data is combined with data developed by the IPPS and is used to provide information regarding the entire shipyard.

The Master Production Schedule (MPS) must be coupled to bills of material structured to support the production process. They are not separate issues. A workable interface between the scheduling and material requirement system is vital. Development of this interface is dependent upon both the planning and material systems employed by the yard. This issue will not be addressed in this paper. However, when a production scheduling system is being developed, the scheduling/materials system interface must be considered.

The MPS also must be supported by engineering. Completion of engineering specifications and drawings must be scheduled to support the production and material ordering process. However, the scheduling of these items will not be considered within the scope of this project.
SYSTEM DEVELOPMENT PHILOSOPHY

The purpose of scheduling is to optimize the use of resources so that the overall production objectives are met. Scheduling involves the assignment of dates to specific tasks. Machine breakdowns, absenteeism, quality and performance problems, material shortages, and other factors complicate the ship building environment. Hence, the assignment of a date does not ensure that the work will be performed at that time. [2] A scheduling system should have the ability to adapt schedules to reflect changes in the ship building environment.

An effective model for use in production scheduling must reflect the strategy by which the ship will be built. These strategies establish the activity durations, resource utilization, and relationships to be used by the Integrated Production Planning System. Documents should be developed to describe the strategy by which the ground assembly, join, outfit, and erect process will occur. Table 1 shows five strategy sheets that, when taken together, will provide the information required to develop an effective Master Production Schedule for the process. (Note: the sheets are illustrated in the build strategy development section of the User’s Manual.) All strategy sheets are reviewed, discussed, and approved prior to model development.

Even if a PC-based model of the production process is not developed, creation of the documents shown in Table 1 is a valuable tool. By bringing together the various materials, engineering, production and support groups for the strategy review process, the build strategies and ship’s design will often be substantially improved and subsequent changes will generally be reduced.

In addition to the strategy sheets, it is advantageous to develop a coding system for the Work Breakdown Structure (WBS) and the Organizational Breakdown Structure (OBS). Use of these codes enables the system to group its output in meaningful ways.
Schedule information is distributed to all required groups in a format meaningful to that group.

<table>
<thead>
<tr>
<th>STRATEGY SHEET</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK BREAKDOWN DIAGRAM</td>
<td>IDENTIFIES BREAKDOWN OF SHIP INTO STRUCTURAL ASSEMBLIES AND SHOWS ASSEMBLIES</td>
</tr>
<tr>
<td></td>
<td>THAT JOIN TOGETHER ('GRAND BLOCK') PRIOR TO ERECTION</td>
</tr>
<tr>
<td>INTEGRATED ASSEMBLY/OUTFIT STRATEGIES</td>
<td>BLOCKS ARE GROUPED BY COMMON TYPE. ASSEMBLY/OUTFIT ACTIVITIES ARE DEFINED.</td>
</tr>
<tr>
<td>BY BLOCK TYPE</td>
<td>DURATIONS, RESOURCE REQUIREMENTS AND GENERAL SCOPE OF WORK FOR EACH ACTIVITY</td>
</tr>
<tr>
<td>GRAND BLOCK STRATEGIES</td>
<td>IDENTIFIED. A SPECIAL CASE OF THE INTEGRATED ASSEMBLY/OUTFIT STRATEGIES.</td>
</tr>
<tr>
<td></td>
<td>IDENTIFIES WHICH BLOCKS ARE JOINED TOGETHER TO FORM GRAND BLOCKS AND THE</td>
</tr>
<tr>
<td></td>
<td>METHOD BY WHICH THEY JOIN.</td>
</tr>
<tr>
<td>ERECTION &quot;STAR&quot; CHART</td>
<td>SHOWS THE DATE EACH ERECTABLE UNIT IS SCHEDULED TO BE JOINED TO THE</td>
</tr>
<tr>
<td></td>
<td>SHIP.</td>
</tr>
<tr>
<td>PROCESS LANE STRATEGIES</td>
<td>SHOWS IN GANTT CHART FORMAT THE FLOW OF BLOCKS THROUGH EACH OF THE</td>
</tr>
<tr>
<td></td>
<td>DEDICATED PROCESS LANES.</td>
</tr>
</tbody>
</table>

Table 1: Ground assembly, join, outfit, and erect strategy sheet descriptions.

A schedule dictates not only the dates on which various activities occur, but also a specific set of material, engineering, facility, and manning requirements. For a schedule to remain credible, it must account for actual material delivery, engineering drawing availability, facility availability, and manpower availability. This cyclic relationship implies that an achievable schedule can be derived only when these factors are considered together.

Data are facts concerning objects, events, relationships, and requirements. Information is data that has been organized in a form that is suitable for decision-making. The development of a
schedule in not an analytically complex task. Development is made complex due to the large volume of data which must be considered. The IPPS transforms the large volume of data which influences schedules into useful information. The clearest way to convey this information is through graphical displays of schedules, manning requirements, and facility utilization data. By showing relevant data in this graphical form, the system serves as a useful tool in generating and updating the MPS.

Based upon the above discussion, the required capabilities of an Integrated Production Planning System can be defined. The requirements for an effective system are shown in table 2.

<table>
<thead>
<tr>
<th>REQUIREMENTS FOR AN INTEGRATED PRODUCTION PLANNING SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ž REFLECT A BUILD STRATEGY</td>
</tr>
<tr>
<td>Ž SHOW MATERIAL, ENGINEERING, FACILITY, AND MANNING REQUIREMENTS IMPLIED BY THE SCHEDULE</td>
</tr>
<tr>
<td>Ž DISPLAY ALL DATA IN A CLEAR MEANINGFUL WAY</td>
</tr>
<tr>
<td>Ž HIGH LEVEL OF FLEXIBILITY AND ADAPTABILITY</td>
</tr>
</tbody>
</table>

Table 2: Production planning system requirements.
SCHEDULE GENERATION SYSTEM OVERVIEW

The flow chart in figure 2 shows the Integrated Production Planning Systems major inputs, outputs, and components. The system consists of four modules which interact to create both the baseline and regularly updated production schedules. The system also generates manning and facility long range and short term requirements. Each module is described in greater detail on the pages that follow.
The New Project Model Generation Module is shown in figure 3. When a new project (i.e. a ship) is brought into the yard, data is gathered regarding build strategies, activity resource requirements, process lane considerations, and block erection data. All this data is collected into a Ship Build Master Data File. This data is passed through the Model Builder Program which creates a Standard Setback Model for the ship.

Figure 3: New Project Model Generation Module.
The Project Integration Module is shown in figure 4. Standard Setback Models take into account build strategies for only the individual ship. Since all ships are built with common facilities and manpower, leveling the MPS is done with all the projects in the yard considered together. The Standard Setback Model is combined with the Standard Setback Models of previously scheduled work to create a Yardwide Schedule Development Model. This model is processed to show the capacity and manpower requirements implied by these schedules. Based upon this information, the model is refined through an iterative process of resource demand leveling. Capacities and manning implied by the schedule are investigated. Schedules are modified until acceptable capacity utilization and manning are achieved. The final iteration of the model is reviewed and approved by the various department heads. Upon approval, this model--becomes the Baseline Production Model.
The Baseline Master Production Schedule Generation Module is shown in figure 5. The Baseline Production Model is used to create the MPS for each project. The model is also used to create and update a database which serves as the baseline schedule for the schedule tracking system. The Baseline Production Model is altered only when a new revision of an existing projects schedule is issued.

From Baseline Production Model

Figure 5: Baseline Master Production Schedule Generation Module
The Production Schedule Update Development Module is shown in figure 6. A copy of the Baseline Production Model is renamed the Production Update Model. This model is updated based on regular meetings and progress data. The updated model is processed and used to generate regularly issued production schedules, manning curves, and facility utilization reports (laydown schedules). The manning curves and facility utilization reports reflect adjustments that have been made from the master schedule to the current production schedule.

From Baseline Production Model

Figure 6: Production Schedule Update Development Module.
PROJECT MANAGEMENT SOFTWARE

The IPPS is built around Welcom Software Technology's Open Plan PC-based project management software. There are several PC-based project management packages on the market today. One of the advantages of Open Plan (herein referred to as the project management software) is that the software package operates within a dBase shell. All of the project management software input and output files are in standard dBase format. This allows all pre-processing and post-processing programs built around the project management software to be written in dBase.

All models shown in the Schedule Generation System Overview exist within the project management software framework. The software takes data regarding individual activities and creates schedules and resource utilization files. The data regarding individual activities is placed into three separate files. The activity file contains the duration of each activity. Before this file is processed, the only dates in the file are the start dates of key events (i.e. the date blocks or grand blocks erect to the ship). The relationship file shows the required sequencing and interaction between various activities. The resource file shows the manning and facility requirements of each activity. The project management software processes these data files and generates all the dates that were not previously defined. These dates are stored to the processed activity file. The software also creates a resource aggregation file. This file shows the utilization of resources as a function of time. The major inputs and outputs of a project management software model are shown in figure 7.

Figure 7: Project management software input and output files.
STEPPING THROUGH THE SYSTEM

To demonstrate how the IPPS is used, schedules were developed and updated for a test case. The test case is the construction of the MV Well Planned, a small, double-hulled product carrier. The first and most important task in scheduling is the development of a build strategy. The build strategy for the MV Well Planned is expressed in terms of the documents described in the Systems Development Philosophy section of this report. The strategy sheets for the MV Well Planned are shown in the Build Strategy Development section of the User’s Manual (pages 6-10).

Information from the documents is used to create the MV Well Planned’s Ship Build Master Data File. This file is processed by the Model Builder Program to create a Standard Setback Model for the ship. This model consists of the activities, resources, and relationships required to assemble, join, and outfit the blocks in preparation for erection. The erection activity for each block is fixed to a particular date as defined by the strategy sheets. Since the final event in each chain of activities is locked, the entire network of activities can be back-scheduled to show the late start and complete dates for each activity in the network.

The Standard Setback Model of the MV Well Planned shows the required start of construction date for the vessel to be 22 weeks before keel. This is not acceptable. To alleviate this early start of construction requirement, the build strategies must be altered. In the case of the MV Well Planned start of construction is driven by the wing tank block assembly process lane. To solve this problem the strategy was altered by using a second build position for this process lane. The revised process lane strategy sheet to reflect this change is shown in figure 8. The model is altered to reflect this new strategy by updating the relationships between the wing tank block assembly activities. The model is then reprocessed. The start of construction date with this new strategy becomes 10 weeks before keel. The build strategy is now
The initial model is back-scheduled to late dates, therefore any leveling done is accomplished by moving activities earlier. Resource leveling strategies must reflect the constraints imposed by a particular yard's capabilities. If a yard has only a limited area to assemble the blocks, schedules may be leveled on space utilization within this area. If there is a required trade for which the yard has limited manning, schedules may be leveled based upon the trade's availability. Schedules may be leveled on any resource or combination of resources included within the model. Since the IPPS operates by back scheduling to late dates, the generalized resource leveling strategy is to first level resources in the area that immediately precedes the erection activity and then work back to earlier activities.

The strategy used for resource leveling of the MV Well Planned is to first level the outfit area manning. Since resources are interchangeable between the pre-blast inverted, pre-bast upright, and post blast outfitting activities, these activities are grouped by their common OBS code and leveled together. Once the outfitting

---

**ASSEMBLY PROCESS LANE (WING TANKS) BLOCK**

<table>
<thead>
<tr>
<th>151</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td></td>
<td></td>
</tr>
<tr>
<td>142</td>
<td></td>
<td></td>
</tr>
<tr>
<td>131</td>
<td></td>
<td></td>
</tr>
<tr>
<td>132</td>
<td></td>
<td></td>
</tr>
<tr>
<td>121</td>
<td></td>
<td></td>
</tr>
<tr>
<td>122</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**KEY**

- ✧ = ERECTION
- ☐ = ADDED FLOAT
- ☐ = PRE-BLAST (UP RIGHT)
- ☐ = PRE-BLAST (INVERTED)
- ☐ = POST BLAST
- ☐ = ASSEMBLY
- ☐ = BLAST & PAINT

Figure 8: Revised process lane strategy sheet.
area is leveled, the assembly area is investigated. The assembly area in this example is leveled based upon the number of blocks with work in progress in both the flat and curved block build areas. By leveling first the outfitting area manning and then the assembly area work in progress, a feasible MPS is created. This schedule reflects the build strategies for the vessel as well as taking into account the manning and facility availability.

Figure 9 shows the output of the Long Range Manning Requirement Generation Program for the outfit area. An analysis of the blocks outfitting during the peak months of April, May, and June show that the majority of the manning requirements during this period of time are driven by the outfitting of the house blocks and house grand blocks. In order to level manning in this area the outfitting of the lower house (grand block 531) will be scheduled prior to the outfitting of the upper house (grand block 533). To change the model to reflect this strategy a single relationship is added. The new relationship forces the outfitting of block 531 to complete before the outfitting of block 533 can start. Since the activities are linked, the system will reschedule the assembly, outfitting, and stacking activities of all blocks which comprise grand block 531. The results of this reschedule (iteration 2) are shown in figure 10.

![Figure 9: Outfit area manning requirements, iteration 1 - by standard setbacks.](image1.png)

![Figure 10: Outfit area manning requirements, iteration 2-grand block 531 forced early.](image2.png)
The manning curve for the outfitting area is now acceptable. Next, an analysis is made of the facility utilization within the assembly area. Two independent resources must be investigated within the assembly area. Both the flat block build platen and the shaped block build platen have limited space. The MPS must be adjusted so as to level both of these resources. Since the resources are independent, they may be leveled simultaneously.

Figures 11 and 12 show the outputs of the Capacity Requirement Generation Program. Note that the system is back-scheduling to late dates and the assembly activity precedes the outfitting activities. Therefore, when assembly activities are forced earlier, float is introduced between the assembly and outfit operations. This has no impact on the outfitting area manning requirements. The resource leveling strategy for the MV Well Planned makes no attempt to level the assembly area manning. However, there is a high correlation between assembly build positions in use and assembly area manning requirements. If assembly build position usage is level, assembly area manning requirements are also fairly level. To level the shaped block assembly platen, some of the blocks scheduled to assemble in April are rescheduled to assemble earlier to fill in the valley in the February–March time period. When leveling the flat block assembly platen it is not desirable to take the excessive work in May and reschedule it for February. This would break the logical build sequence for the ship. Instead, the schedule should be modified to push earlier the building of a few blocks in March, April, and May.
This will eliminate the excessive capacity requirements while maintaining a proper build sequence. The results of this rescheduling (iteration 3) are shown in figure 13 and 14.

The facility utilization within the assembly area is now acceptable. This model is named the Baseline Production Model and processed by the Master Schedule Generation Program to create a Master Production Schedule. The MPS is approved by production, engineering, materials and support groups and the schedule is issued. An upload file is created to support the schedule tracking system.

A copy of the Baseline Production Schedule is renamed the Production Update Model. This model is updated based upon regular meeting and progress data. These weekly meetings are attended by members of the assembly, outfitting, erection, and support groups. These meetings serve to update the short term schedule documents based upon actual and projected progress.
Figure 15 shows the current production schedule for the flat platen assembly area. At the production update meeting, the assembly area representative will report on actual and projected progress. The assembly area laydown chart is marked up by the assembly area representative as shown in figure 16. Based on the assembly area inputs, changes will be made to the production schedule. These changes are shown in table 3. The Production Update Model is modified to reflect the actions taken in the meeting. The model is then reprocessed and updated production schedules are issued to appropriate groups.

<table>
<thead>
<tr>
<th>BUILD POSITION</th>
<th>NOV</th>
<th>DEC</th>
<th>JAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-01</td>
<td>BLK141</td>
<td>BLK131</td>
<td></td>
</tr>
<tr>
<td>01-02</td>
<td>BLK241</td>
<td>BLK231</td>
<td>BLK221</td>
</tr>
<tr>
<td>01-03</td>
<td>BLK311</td>
<td>BLK322</td>
<td>BLK341</td>
</tr>
<tr>
<td>01-04</td>
<td>BLK242</td>
<td>BLK222</td>
<td></td>
</tr>
</tbody>
</table>

Figure 16: Flat platen assembly laydown charts modified by the assembly area rep. at the production update meeting.

<table>
<thead>
<tr>
<th>CONFLICT</th>
<th>ADJUSTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK 2421 IN BUILD POSITION 1-4 WILL OVERLAP</td>
<td>ASSEMBLY AGREES TO CRASH THE DURATION OF BLOCK 232 IN ORDER TO RECOVER TO THE SCHEDULE.</td>
</tr>
<tr>
<td>WITH LAYDOWN OF BLOCK 232.</td>
<td>OUTFITTING, MADE AWARE OF THE DELAY AND ITS IMPACT AHEAD OF TIME, AGREES TO WORK THIS BLOCK MORE AGGRESSIVELY TO MAKE UP FOR THE DECREASED DURATION.</td>
</tr>
<tr>
<td>THE EXTENDED DURATION OF BLOCK 311 IN BUILD POSITION 1-3 DOES NO CAUSE A CONFLICT IN THE ASSEMBLY AREA. HOWEVER, THIS 3 DAY DELAY WILL CUT INTO THE SCHEDULED OUTFITTING DURATION,</td>
<td>LATER ACTIVITIES ARE NOT AFFECTED, THEREFORE, THIS DELAY HAS NO IMPACT.</td>
</tr>
<tr>
<td>THE EXTENDED DURATION OF BLOCK 241 IN BUILD POSITION 1-2 HAS NO IMPACT ON OTHER BLOCKS IN THE ASSEMBLY AREA. THERE IS FLOAT BETWEEN THIS ACTIVITY AND Outfitting ACTIVITIES OF THIS BLOCK</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Adjustments made to the production schedule.
CONCLUSION

An effective MPS must reflect not only a build strategy, but also material, engineering, facility, and manpower availability. A PC-based system can be established to assist in creating and updating the MPS.

The system described in this reference will allow shipyard personnel to see the production schedule in a clear, meaningful way. This system can serve as a tool to help a shipyard make better production decisions and operate with improved effectiveness.

REFERENCES


# USER’S MANUAL

## INTEGRATED PRODUCTION PLANNING SYSTEM

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM SET UP</td>
<td></td>
</tr>
<tr>
<td>SOFTWARE/HARDWARE REQUIREMENTS</td>
<td>2</td>
</tr>
<tr>
<td>SOFTWARE INSTALLATION</td>
<td>3</td>
</tr>
<tr>
<td>SYSTEM USE</td>
<td></td>
</tr>
<tr>
<td>SYSTEM INFORMATION REQUIREMENTS</td>
<td>4-13</td>
</tr>
<tr>
<td>PROJECT CREATION</td>
<td>13-23</td>
</tr>
<tr>
<td>SYSTEM'S FILE REQUIREMENTS AND STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>MODEL BUILDER PROGRAM</td>
<td></td>
</tr>
<tr>
<td>MODEL BUILDER DATA FILE</td>
<td></td>
</tr>
<tr>
<td>MODEL CREATION</td>
<td></td>
</tr>
<tr>
<td>SCHEDULE DEVELOPMENT</td>
<td>24-30</td>
</tr>
<tr>
<td>PROCESSING THE MODEL</td>
<td></td>
</tr>
<tr>
<td>DEVELOPING THE LAYDOWN CHARTS</td>
<td></td>
</tr>
<tr>
<td>DEVELOPING RESOURCE UTILIZATION CURVES</td>
<td></td>
</tr>
<tr>
<td>DEVELOPING GANTT CHARTS</td>
<td></td>
</tr>
<tr>
<td>PROGRESSING AND UPDATING THE MODEL</td>
<td>31</td>
</tr>
<tr>
<td>CUSTOMIZING THE SYSTEM</td>
<td>31</td>
</tr>
<tr>
<td>HELP</td>
<td>32</td>
</tr>
</tbody>
</table>
This User’s Manual is provided as a reference for shipyards wishing to develop a PC-based Integrated Production Planning System (IPPS). The software included in this package is not intended to be a turnkey system. To effectively implement a system, a shipyard should establish an IPPS development team. The team will define what that particular yard’s needs are in an IPPS. They will also compile the ship’s build strategy in proper format for use by the system.

This manual assumes that the development team has a moderate knowledge of dBase programming and an understanding of project management techniques and ship production planning. The user’s group must also become familiar with the Open Plan software system.

The coding included with this package will operate for the sample organization and ship described herein. This package is intended as a skeletal outline showing how such a system can be established. For an IPPS to work at a particular shipyard, the yard must modify the coding so that the system will conform to the yard’s facilities and method of operation. The coding and documentation provided is a base from which customized programs can be written for a yard’s specific application. In developing and using the IPPS, the development team needs to have access to the following:

- This User’s Manual
- An MS-DOS System
- Open Plan User’s Manuals
- The Software and Hardware Described in the System Set-Up Section Which Follows
SYSTEM SET-UP

software/Hardware Requirements

The IPPS is developed to run on IBM compatible MS-DOS systems. The data files created when using this system will contain thousands to tens of thousands of records. For ease of use, a high performance 386 or 486 machine is recommended so that the programs will run in a reasonable time frame. The system as laid out in this report requires three pieces of commercially available software. The project management software selected is Open Plan. Open Plan operates within a dBase or FoxPro environment, so one of these pieces of software is also required. All coding included within this package is written in dBase. Manning curves generated by the system can be created by any graphics program that can chart an imported dBase file. The system described in this report makes use of Lotus Freelance. Points of contact for required software are shown in table 1.

<table>
<thead>
<tr>
<th>PRODUCT</th>
<th>DEVELOPER</th>
<th>POINT OF CONTACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Plan</td>
<td>Welcom Software</td>
<td>WST corporation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15995 North Barkers Landing, Suite 275</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Huston, Tx77079</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tel: (713)558-0514</td>
</tr>
<tr>
<td>dBasee III</td>
<td>Borland International</td>
<td>Local software dealer</td>
</tr>
<tr>
<td>Fox Pro</td>
<td>Fox Software</td>
<td>Local software dealer</td>
</tr>
<tr>
<td>Freelance</td>
<td>Lotus Development</td>
<td>Local software dealer</td>
</tr>
</tbody>
</table>

Table 1: Software requirements for the IPPS,
Software Installation Procedure

To install the IPPS programs and files onto your hard drive:

1) Open Plan must be installed and running on your computer.
2) Insert "Program Disk" Disk 1 (included in this project) into Drive "B"
3) Use the DOS COPY Command to copy all files from the "Program Disk1" Disk 1 into the Open Plan Directory. This is done by typing the statement shown below after prompt.
   C:> COPY B:,*.* C:\OPLAN
4) Repeat the copy process for the "Sample Projects" Disk 2 (included in this package). Open Plan usually has a separate sub-directory for each of its projects, so make sure you enter the proper path. This is done by typing the statement shown below after the prompt.
   C:> COPY B:,*.* C:\OPLAN\PROJ
5) When in Open Plan you must add the two sample projects, "MODEL" and "MODE", to Open Plan's Project Directory. Then use the Re-Build Index Utility to re-index the project directory and both projects.

Figure 1 lists all of the files copied from both the Program Disk and Sample Projects Disk.

Note: MODEL is the empty project with all necessary changes to the file structures. MODELA is a completely built project with table assignments for area laydown requirements. This model was created by processing the Build.dbf data file through the Model Builder Program. The standard setback model is then level loaded by both area and manning utilization.

Figure 1: Installing the Integrated Production Planning System.
SYSTEM USE

System Information Requirements

An effective model for use in production scheduling must reflect the strategy by which the ship will be built. This section will discuss what information should be developed and illustrates sample formats for gathering this information.

Documents may be developed to describe the strategy by which the ground assembly, join, outfit, and erect process will occur. Table 1 of the Written Report shows five strategy sheets that, when taken together, will provide the information required to develop an effective MPS for the process. Figure 2 below shows the strategy development logic flowchart for the creation of the Model Builder Data Base.

![Strategy development logic flow](image)

Figure 2: Strategy development logic flow.
The first step in developing this information is to create the block/grand block breakdown for the ship. This breakdown (shown in figure 3) drives the pre-outfitting strategy for the vessel. Once the block/grand block breakdown has been established, the other build strategy sheets may be developed. The ground assembly, join, outfit, and erect strategy sheets are shown in figures 4 through 7. An effective build strategy is a function of a yards facilities, capacities, and ship design. These documents should be developed through the joint effort of production, planning, materials, and engineering. This development should be done concurrently with the development of structural drawings and composites so that these documents can both influence and be influenced by the development of the block breakdown and build strategy.
Figure 3: Block/Grand Block Breakdown for the MV WELL PLANNED. This figure shows the manner in which the vessel will be broken down into modules for construction. The figure also shows which of these modular blocks are joined together or "Grand Blocked" prior to their erection onto the ship. Each Block and Grand Block is assigned a unique name (a 3 digit number) so that the blocks may be identified and scheduled by the Integrated Production Planning System.
Figure 4: Erection "star" chart for the MV WELL PLANNED.

This figure shows the dates that each erectable block and grand block will be brought to the ship. These dates are the only ones that are initially fixed within the Integrated Production Planning System.

All other dates (including the stack dates of blocks that go into grand blocks) are developed through the IPPS.
INTEGRATED ASSEMBLY/OUTFIT STRATEGIES
BY BLOCK TYPE

FIGURE 5: Integrated Assembly/Outfit Strategies for the MV WELL PLANNED. This figure shows the build strategy activity durations in days, and manning requirements in hours for each of the block making up the ship. This table is developed jointly between steel and outfit planners and production personnel. Durations and budgets shown are sufficient to accomplish both the structural and the outfitting work planned in each step. Work should be scheduled so that the optimum amount of outfitting is installed on the block prior to erection. Strategies should also consider the sequencing of structural and outfitting work so that they proceed in an effective manner.

<table>
<thead>
<tr>
<th>BLK TYPE</th>
<th>BLOCK NUMBERS</th>
<th>BUILD STRAT.</th>
<th>ASSY</th>
<th>PRE-BLAST INVERTED</th>
<th>PRE-BLAST UP RIGHT</th>
<th>BLAST &amp; PAINT</th>
<th>POST BLAST</th>
<th>EFFECT STACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOUBLE BOTTOM</td>
<td>020,030 040,050</td>
<td>STANDARD</td>
<td>20 1000</td>
<td>-</td>
<td>5 250</td>
<td>5 100</td>
<td>5 250</td>
<td>20 1000</td>
</tr>
<tr>
<td>TRANS. BHD.</td>
<td>123,133 143</td>
<td>STANDARD</td>
<td>15 800 5 150</td>
<td>-</td>
<td>0 0</td>
<td>0 0</td>
<td>20 400</td>
<td></td>
</tr>
<tr>
<td>STERN</td>
<td>160,281 282</td>
<td>STANDARD</td>
<td>15 2000 10 300 5 150</td>
<td>5 100</td>
<td>5 150</td>
<td>20 1500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stern</td>
<td>181,162</td>
<td>GRAND2</td>
<td>15 1000</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20 200</td>
<td></td>
</tr>
<tr>
<td>MAIN DECK</td>
<td>261,262</td>
<td>STANDARD</td>
<td>16 800 5 200</td>
<td>-</td>
<td>6 100</td>
<td>5 200</td>
<td>20 600</td>
<td></td>
</tr>
<tr>
<td>MAIN DECK</td>
<td>221-&gt;242</td>
<td>GRAND4</td>
<td>16 1000</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20 150</td>
<td></td>
</tr>
<tr>
<td>WING</td>
<td>121-&gt;152</td>
<td>STANDARD</td>
<td>15 800 5 150</td>
<td>-</td>
<td>10 150</td>
<td>5 150</td>
<td>20 600</td>
<td></td>
</tr>
<tr>
<td>BOW</td>
<td>200</td>
<td>STANDARD</td>
<td>20 1500 5 125 6 125</td>
<td>5 100</td>
<td>10 125</td>
<td>20 1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOW</td>
<td>010,100</td>
<td>GRAND3</td>
<td>20 2000</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20 300</td>
<td></td>
</tr>
<tr>
<td>HOUSE</td>
<td>310-&gt;350</td>
<td>GRAND1</td>
<td>15 800 20 1000</td>
<td>5 200</td>
<td>5 100</td>
<td>10 200</td>
<td>20 300</td>
<td></td>
</tr>
<tr>
<td>MAST</td>
<td>401,402</td>
<td>SHOP</td>
<td>- 10 300</td>
<td>-</td>
<td>5 100</td>
<td>700</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>RUDDER</td>
<td>499</td>
<td>STANDARD</td>
<td>30 1000</td>
<td>-</td>
<td>-</td>
<td>5 100</td>
<td>500</td>
<td>20 650</td>
</tr>
<tr>
<td>CASEING</td>
<td>380</td>
<td>STANDARD</td>
<td>16 900 10 1000</td>
<td>6 500</td>
<td>10 500</td>
<td>20 650</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STACK</td>
<td>400</td>
<td>STANDARD</td>
<td>20 1000</td>
<td>- 10 500 5 500</td>
<td>10 20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 5: Integrated Assembly/Outfit Strategies for the MV WELL PLANNED. This figure shows the build strategy activity durations in days, and manning requirements in hours for each of the block making up the ship. This table is developed jointly between steel and outfit planners and production personnel. Durations and budgets shown are sufficient to accomplish both the structural and the outfitting work planned in each step. Work should be scheduled so that the optimum amount of outfitting is installed on the block prior to erection. Strategies should also consider the sequencing of structural and outfitting work so that they proceed in an effective manner.
Figure 6: Grand Block Strategy for the lower house of the MV WELL PLANNED. This figure shows the build strategy, activity durations, and manning requirements for the lower house (GB 531). The figure shows which blocks comprise this grand block and the time phasing by which these blocks stack into the grand block. A sheet such as this is prepared for each of the vessel’s grand blocks.
Figure 7: Process Lane Strategy for the MV WELL PLANNED. The figure shows a Gantt chart for production of wing tanks. The assembly process for each of these blocks will occur in a single dedicated work position. In order to accommodate this, float is introduced into the schedule of the early blocks through the process lane.
As noted in the System Development Philosophy section of the Project Report, a Work Breakdown Structure (WBS) and Organizational Breakdown Structure (OBS) should be created so that all data developed by the system may be viewed and aggregated in meaningful ways. These structures are shown in figures 8 and 9.

**WORK BREAKDOWN STRUCTURE**

```
YARD WIDE PRODUCTION
       \   /
       1   2
         \ /  \
      SHEP 1 SHEP 2 SHEP N
       \   /
       11  12 1N
          \  /  \
         BLK 0001 BLK 0002 BLK XXXX
         \   /
        110001 110002 11XXXX
          \  /
         STEEL NON-STEEL
           /  /
         1100011 1100012
            /  /  \
           FABRICATION SUB ASSEMBLY ASSEMBLY ERECT/JION
               /  /  /  /
           11000111 11000112 11000113 11000114
                         /  /  /  \
                       PIPE SHOP VENT SHOP OUTFIT BLAST & PAINT
                           /  /  /  /
                       11000121 11000122 11000123 11000124
                           /  /  /  \
                          FLAT CURVED ERECT JOIN PRE BLAST POST BLAST
                           /   /  /  /  /
                          110001131 110001132 110001141 110001142 110001231 110001232
                          /  /  /  /  /
                         INVERTED UPRIGHT
                           1100012311 1100012312
```

Figure 8: WBS for the MV WELL PLANNED sample.
In addition to build strategy, WBS, and OBS it is also necessary to gather information regarding resource requirements. Manning, critical materials, crane requirements, and space utilization are just a few items that can be defined as resources. Schedule refinement is an attempt to balance utilization of resources while maintaining the build strategy and meeting your project objectives. Therefore, selecting which resources to track is a key decision in the development of an Integrated Production Planning System. Resources to be tracked by the system should be those considered critical and which the yard wishes to control as Master Production Schedules are developed. The decision of which resources to track and how to code these resources should be made in the initial stages of the IPPS development. The decision will
determine the contents of the Resource Library File, the structure of the Curve Generation File, and the coding of the Model Builder and Curve Generation Programs (these files and programs will be explained later in this manual). The resources tracked by the IPPS are shown in Table 2.

<table>
<thead>
<tr>
<th>RESOURCE</th>
<th>RESOURCE CODE</th>
<th>UNIT OF MEASURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEELFABRICATION HOURS</td>
<td>H111</td>
<td>HOURS</td>
</tr>
<tr>
<td>STEEL SUB-ASSEMBLY HOURS</td>
<td>H112</td>
<td>HOURS</td>
</tr>
<tr>
<td>STEEL ASSEMBLY HOURS</td>
<td>H113</td>
<td>HOURS</td>
</tr>
<tr>
<td>BLAST &amp; PAINT HOURS</td>
<td>H124</td>
<td>HOURS</td>
</tr>
<tr>
<td>OUTFITTING HOURS</td>
<td>H123</td>
<td>HOURS</td>
</tr>
<tr>
<td>ERECTION HOURS</td>
<td>H114</td>
<td>HOURS</td>
</tr>
<tr>
<td>FLAT ASSEMBLY POSITION</td>
<td>P1131</td>
<td>EACH</td>
</tr>
<tr>
<td>CURVED ASSEMBLY POSITION</td>
<td>P1132</td>
<td>EACH</td>
</tr>
<tr>
<td>BLAST &amp; PAINT POSITION</td>
<td>P124</td>
<td>EACH</td>
</tr>
<tr>
<td>ON-BLOCK INVERTED O/F POSITION</td>
<td>P12311</td>
<td>EACH</td>
</tr>
<tr>
<td>ON-BLOCK UPRIGHT O/F POSITION</td>
<td>P12312</td>
<td>EACH</td>
</tr>
<tr>
<td>ON-BLOCK POST BLAST O/F POSITION</td>
<td>P1232</td>
<td>EACH</td>
</tr>
<tr>
<td>ASSEMBLY COMPLETE</td>
<td>P-A</td>
<td>EACH</td>
</tr>
<tr>
<td>BLOCK ERECTION/STACKING</td>
<td>P-E</td>
<td>EACH</td>
</tr>
</tbody>
</table>

Table 2: Critical resources tracked for the MV WELL PLANNED.

Project Creation

This section will describe the required format in which the production strategy information is entered for use by the IPPS. All information encompassed by the IPPS is written to and calculated from dBase files. The files used by the system and a brief explanation of how these files and structures are developed are shown in table 3. A full listing of all file structures and their required format is found in appendix I. The IPPS uses custom programs written for this project and the Open Plan software system. This User’s Manual attempts to explain those items not covered in the Open Plan User’s Manual. In developing and using an IPPS based upon this report, the user should refer to both this manual and that provided by Open Plan.
<table>
<thead>
<tr>
<th>FILE</th>
<th>DESCRIPTION</th>
<th>dBASE STRUCTURE CREATION</th>
<th>dBASE RECORD CREATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILD DATA</td>
<td>Used by model builder program to create the records of the activity, resource, and relationship file.</td>
<td>Created by system user.</td>
<td>Created by system user from strategy sheets.</td>
</tr>
<tr>
<td>PROCESS LANE DATA</td>
<td>Used by model builder program to establish block to block process into relationships.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WBS CODE</td>
<td>Defines the work breakdown structure to be used to sort and group information in meaningful ways.</td>
<td>Created by system user.</td>
<td></td>
</tr>
<tr>
<td>OBS CODE</td>
<td>Defines the organizational breakdown structure to be used to sort and group information in meaningful ways.</td>
<td>Structure defined interactively through Open Plan.</td>
<td>Created by user interaction through Open Plan.</td>
</tr>
<tr>
<td>GRAND BLOCK CODE</td>
<td>Defines the erectable unit to which each block will go.</td>
<td>Structure defined interactively through Open Plan.</td>
<td>Created by model builder program.</td>
</tr>
<tr>
<td>CALENDER</td>
<td>Defines work period, how many days per week worked.</td>
<td>Created by Open Plan</td>
<td>Developed interactively through Open Plan menu system.</td>
</tr>
<tr>
<td>HOLIDAY</td>
<td>Links with calendar file to define days off.</td>
<td>Created by Open Plan</td>
<td>Developed interactively through Open Plan menu system.</td>
</tr>
<tr>
<td>RESOURCE LIBRARY</td>
<td>Defines resources to be tracked by the system.</td>
<td>Created by Open Plan</td>
<td>Created by Open Plan through Open Plan menu system.</td>
</tr>
<tr>
<td>ACTIVITY</td>
<td>Defines the production activities to be modeled by the system.</td>
<td>Created by Open Plan, requires some modification from standard structure.</td>
<td>Created by model builder based upon info. contained in the Model Data Fllo.</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>Defines the resource requirements for each activity</td>
<td>Created by Open Plan, requires some modification from standard structure.</td>
<td>Created by model builder based upon info. contained in the Model Data Fllo.</td>
</tr>
<tr>
<td>RELATIONSHIP</td>
<td>Defines the relationship between the production activities.</td>
<td>Created by Open Plan, requires some modification from standard structure.</td>
<td>Created by model builder based upon info. contained in the Model Data Fllo.</td>
</tr>
<tr>
<td>AGGREGATION</td>
<td>Defines the resource requirements aggregated as a function of time.</td>
<td>Created by Open Plan</td>
<td>Generated through Open Plan processing.</td>
</tr>
<tr>
<td>CURVE GENERATION</td>
<td>Rewrites the resource aggregation file in a form compatible with Freelance for curve generation.</td>
<td>Created by system user.</td>
<td>Generated by the Curvo Generation Program using the Aggregation Fllo.</td>
</tr>
</tbody>
</table>

Table 3: Files required for use by the Integrated Production Planning System.
The Integrated Production Planning System models the activities associated with the assembly, outfitting, joining, and erection of hull blocks. A scheduling model with sufficient detail to meet the system objectives will be large. A model created for a 700 foot container vessel consists of approximately 3000 activities, 3000 relationships, and 10,000 resource requirements. These numbers will vary depending upon the size and complexity of the vessel and the sizes of the interim products. A yardwide integrated planning model is too large to make practical the entering of all relevant data by keyboard. Therefore, a program has been developed to build a standard setback model for a ship. This Model Builder Program uses generic strategies by block type and block specific data. Each build strategy establishes a set of activities and sequence by which a block is assembled, joined, outfitted and erected. Build strategies must be established that describes the manner in which all blocks to be tracked by the IPPS will be constructed. These build strategies are hard coded into the Model Builder Program. The build strategies used for the MV WELL PLANNED hard coded into the Model Builder Program are shown in figure 10. The Model Builder Program obtains the block specific information from the Build Data File. In addition to the Build Data File, a Process Lane Data File is created. This file establishes the link between a process lane activity and the predecessor block’s process lane activity. All information entered into these files can be found on the strategy sheets. The information required by the Build Data File and Process Lane Data File, along with the strategy sheets from which this information is obtained, is shown in tables 4 and 5. The Model Builder Program creates activity, resource, and relationship files in proper format to be used by the project management software. For the detailed explanation of how the program performs this task, see the program logic flowchart in appendix IV.
Figure 10: Strategies hard coded into the Model Builder Program for the MV WELL PLANNED.
"Assembly Stack" Build Strategy (GRAND2)

Fabrication ➔ Sub-Assembly ➔ Assembly ➔ Stacking ➔ Pre-Blast O/F (Inverted) ➔ Pre-Blast O/F (Upright) ➔ Blast & Paint ➔ Post-Blast Outfitting ➔ Erection

For each "Stacking Block" [Bik< 600]

For "Grand Block" [Bik≥ 600]

"Pre-Blast (Upright) Stack" Build Strategy (GRAND3)

Fabrication ➔ Sub-Assembly ➔ Assembly ➔ Stacking ➔ Pre-Blast O/F (Upright) ➔ Blast & Paint ➔ Post-Blast Outfitting ➔ Erection

For each "Stacking Block" [Bik< 600]

For "Grand Block" [Bik≥ 600]

"Pre-Blast (Inverted) Stack" Build Strategy (GRAND4)

Fabrication ➔ Sub-Assembly ➔ Assembly ➔ Stacking ➔ Pre-Blast O/F (Inverted) ➔ Pre-Blast O/F (Upright) ➔ Blast & Paint ➔ Post-Blast Outfitting ➔ Erection

For each "Stacking Block" [Bik< 600]

For "Grand Block" [Bik≥ 600]

Figure 10: (Continued) Strategies hard coded into the Model Builder Program for the MV WELL PLANNED.
<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>DESCRIPTION</th>
<th>INPUT FROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION</td>
<td>Block number</td>
<td>Integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BLK_TYP</td>
<td>Block typo</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>STRATEGY</td>
<td>Block build strategy</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>GRAND</td>
<td>Grand block to which each block stacks</td>
<td>Grand Block Strategy Shoot</td>
</tr>
<tr>
<td>ERECTION</td>
<td>Data each block erects to ship (if block stacks to grand block, field will be blank.)</td>
<td>Erection 'Star' Chart</td>
</tr>
<tr>
<td>LAG_E</td>
<td>Duration from the day the first stacking block joins the grandblock to the day this block joins the grandblock.</td>
<td>Grand Block Strategy Sheet</td>
</tr>
<tr>
<td>ASSY_POS</td>
<td>Assembly build position requirement, flat or curved build platten, to be determined from block type.</td>
<td>Integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>ASSY-D</td>
<td>Assembly duration</td>
<td>Integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>OF1_D</td>
<td>Pro-blast inverted outfit duration</td>
<td>Integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>OF2_D</td>
<td>Pre-blast upright outfit duration</td>
<td>integrated Assembly/Outfit Strat Sheet</td>
</tr>
<tr>
<td>OF3-D</td>
<td>Blast &amp; paint duration</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>OF4_D</td>
<td>Post-blast outfit duration</td>
<td>integrated Assembly/Outfit Strat Sheet</td>
</tr>
<tr>
<td>BUD_OF1</td>
<td>Pro-blast inverted outfit budget</td>
<td>Integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BUD_OF2</td>
<td>Pro-blast upright outfit budget</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BUD_OF3</td>
<td>Blast &amp; paint budget</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BUD_OF4</td>
<td>Post-blast outfit budget</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BUD_FAB</td>
<td>Fabrication budget</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BUD_SUB</td>
<td>Sub-assembly budget</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BUD_ASSY</td>
<td>Assembly budget</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>BUD_ERECT</td>
<td>Erection budget</td>
<td>integrated Assembly/Outfit Strat Shoot</td>
</tr>
<tr>
<td>FIRST_S</td>
<td>If block stocks to grand block, identifies the first block to stock to the grand block.</td>
<td>Grand Block Strategy Shoot</td>
</tr>
</tbody>
</table>

Table 4: Data fields of the Build Data File.
<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>DESCRIPTION</th>
<th>INPUT FROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Activity identification for the stage/sub-stage of the block going through the process lane.</td>
<td>Process Lane Strategy Shoot</td>
</tr>
<tr>
<td>PRED</td>
<td>Activity identification for the block proceeding the block listed in ID through the process lane.</td>
<td>Process Lane Strategy Shoot</td>
</tr>
<tr>
<td>TYPE</td>
<td>Type of relationship between activities. Since only one block may be in a dedicated work position at a time, a finish to start (FS) relationship will be used.</td>
<td></td>
</tr>
<tr>
<td>LAG</td>
<td>The time interval between the activity and its predecessor. If you want one activity to start the day after the other is completed, the lag is 0.</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Data fields of the Process Lane Data File.

Note: Appendix ii contains a set of strategy sheets identifying specifically where the data for each field of the Build Data File and Process Lane Data File is located.
Once the Build Data File and the Process Lane Data File have been created, the Model Builder Program can be run. The program creates a series of activities, resources, and relationships for each block from the build strategies and the Build Data File. Inter-block process lane relationships are established from information in the Process Lane Data File. ID's for each activity created by the Model Builder Program are established based upon the contract letter, block number, and the stage of operation for each block. The ID coding system used by the Model Builder Program is shown in figure 11.

![Figure 11: Activity ID as created by the Model Builder Program.](image)

Figure 12 illustrates the activities, resources, relationships, and codes created by the Model Builder Program for a block of standard build strategy. Table 6 shows the sequence of steps necessary to build a setback model for a vessel using the IPPS.
Figure 12: Model as created by the Model Builder Program for a block of standard strategy on the MV WELL PLANNED.
1. Create an empty Open Plan project as explained in the Open Plan manuals (e.g. WELLPLAN).

2. Create the structures for the code files through Open Plan.
   - Code File 1 (C1) - WBS
   - Code File 2 (C2) - OBS
   - Code File 4 (C4) - Grand Block Code

3. Create the Calendar File and Holiday File through Open Plan.

4. Build the Resource Library and the Resource Availability files (e.g. OVERALL) through Open Plan showing the resources the system is to track as in table 2. Create the structure of the Curve Generation File (e.g. CURVES) as shown in table 6A.

5. Build the OBS file (Code C2) through Open Plan. A sample of the OBS file created for the MV WELL PLANNED is shown in Appendix III.

   Note: A common Calendar File, Holiday File, OBS File, and Resource Library File should be used for all projects being tracked by the IPPS.

6. Generate a setback model through the Model Builder Program.
   - Select the Integrated Plan System menu choice under the Process heading of Open Plan’s main menu.
   - Select menu option 1: New Model Generation.
   - Select menu option 1: Model Builder Program.
   - Enter the contract letter designator of the ship for which you are building the model (you must have a separate contract letter for each vessel being modeled by the IPPS).
   - Enter the project name corresponding to the project name created in step (1) along with the path to which you want the data files created by the IPPS written. This path is the path to which Open Plan writes each of its projects (e.g. c:\WELLPLAN).
   - Enter the name of the Build Data Base with path (e.g. c:\BUILD).

   The Model Builder Program creates the activity, resource, and relationship file in Open Plan for each block and grand block. If the Model Builder Program runs successfully, it returns a message saying the project has been created and asks if you wish to add any process lane relationships.
   - Enter the field name and path of the Process Lane Data File (e.g. c:\P_LANE).
   - This adds the block to block process lane relationships.
   - The system then asks if you wish to build the WBS code file (C1).
   - Enter ‘Y’.
   - Enter the WBS code file path and name as created in step (2) (e.g. c:\SP8_c1).
   - This creates the C1 data file. The system then asks if you wish to create a new grand block code field (C4).
   - Enter ‘Y’.
   - Enter the Grand Block Code File path and name as created in step (2) (e.g. c:\SP8_C4).
   - Escape back to the Open Plan main menu.

7. Reindex the model by selecting the Rebuild Indexes menu choice under the System menu heading of Open Plan’s main menu. When in the Rebuild Index facility of Open Plan select 1 and 2 and then 9 to exit.

Table 6: Model creation sequence of steps.
<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODATE</td>
<td>LAST DAY OF EACH AGGREGATION PERIOD</td>
</tr>
<tr>
<td>WORKPDS</td>
<td>NO. OF WORK DAYS SCHEDULED IN AGG. PERIOD</td>
</tr>
<tr>
<td>H111</td>
<td>STEEL FABRICATION HOURS</td>
</tr>
<tr>
<td>H112</td>
<td>STEEL SUB-ASSEMBLY HOURS</td>
</tr>
<tr>
<td>H113</td>
<td>STEEL ASSEMBLY HOURS</td>
</tr>
<tr>
<td>H1141</td>
<td>STEEL ERECT HOURS</td>
</tr>
<tr>
<td>H1142</td>
<td>STEEL JOIN HOURS</td>
</tr>
<tr>
<td>H114</td>
<td>STEEL JOIN/ERECT HOURS</td>
</tr>
<tr>
<td>H12311</td>
<td>PRE-BLAST INVERTED OUTFIT HOURS</td>
</tr>
<tr>
<td>H12312</td>
<td>PRE-BLAST UPRIGHT OUTFIT HOURS</td>
</tr>
<tr>
<td>H1231</td>
<td>PRE-BLAST OUTFIT HOURS</td>
</tr>
<tr>
<td>H1232</td>
<td>POST-BLAST OUTFIT HOURS</td>
</tr>
<tr>
<td>H123</td>
<td>TOTAL OUTFIT HOURS</td>
</tr>
<tr>
<td>H124</td>
<td>BLAST &amp; PAINT HOURS</td>
</tr>
<tr>
<td>H11</td>
<td>TOTAL STEEL HOURS</td>
</tr>
<tr>
<td>H12</td>
<td>TOTAL OUTFIT/BLAST &amp; PAINT HOURS</td>
</tr>
<tr>
<td>H1</td>
<td>TOTAL HOURS</td>
</tr>
<tr>
<td>P_A</td>
<td>BLOCK ASSEMBLIES COMPLETE</td>
</tr>
<tr>
<td>P_E</td>
<td>BLOCK ERECTIONS COMPLETE</td>
</tr>
<tr>
<td>P1131</td>
<td>FLAT ASSEMBLY POSITIONS IN USE</td>
</tr>
<tr>
<td>P1132</td>
<td>CURVED ASSEMBLY POSITIONS IN USE</td>
</tr>
<tr>
<td>P113</td>
<td>TOTAL ASSEMBLY POSITIONS IN USE</td>
</tr>
<tr>
<td>P12311</td>
<td>PRE-BLAST INV. OUTFIT POSITIONS IN USE</td>
</tr>
<tr>
<td>P12312</td>
<td>PRE-BLAST UPRIGHT OUTFIT POSITIONS IN USE</td>
</tr>
<tr>
<td>P1231</td>
<td>PRE-BLAST OUTFIT POSITIONS IN USE</td>
</tr>
<tr>
<td>P1232</td>
<td>POST-BLAST OUTFIT POSITIONS IN USE</td>
</tr>
<tr>
<td>P123</td>
<td>TOTAL OUTFIT POSITIONS IN USE</td>
</tr>
<tr>
<td>P124</td>
<td>BLAST &amp; PAINT POSITIONS IN USE</td>
</tr>
<tr>
<td>P12</td>
<td>TOTAL O/F AND B &amp; P POSITIONS IN USE</td>
</tr>
</tbody>
</table>

Table 6A Structure of the Curve Generation File for the MV WELL PLANNED.
Schedule Development

The steps described in table 6 will create a model for a single hull. However, the IPPS tracks the ground assembly, outfit, joining, and erection of all projects in process and planned for the yard. It is therefore necessary to merge the model created by the Model Builder Program (e.g. WELLPLAN) with a single, yardwide Open Plan project (e.g. OVERALL). This model contains all the work to be tracked by the IPPS on current and planned contracts. The project OVERALL has the same Calendar File, Holiday File, OBS file, and Resource Library File as that of the individual vessel models. The WBS and Grand Block Code Files for OVERALL are a combination of the WBS and Grand Block Code Files of the individual vessels being tracked. To merge the project WELLPLAN into the project OVERALL, follow the sequence of steps shown in table 7.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Select the Merge Copy menu choice under the Utilities heading of Open Plan’s main menu.</td>
</tr>
<tr>
<td>(2)</td>
<td>Choose menu option 2: Merge</td>
</tr>
<tr>
<td></td>
<td>1 Enter the project to contain the merged projects (e.g. OVERALL).</td>
</tr>
<tr>
<td></td>
<td>1 The program asks if you want to delete all project data. Enter “No”.</td>
</tr>
<tr>
<td></td>
<td>1 Enter the project you are merging into the model (e.g. WELLPLAN).</td>
</tr>
<tr>
<td></td>
<td>1 The program asks if you need to renumber. Enter ‘No’.</td>
</tr>
<tr>
<td></td>
<td>1 The program asks if you wish to merge another project. Leave blank, hit return.</td>
</tr>
<tr>
<td>(3)</td>
<td>Enter ‘9’ to return to the Open Plan main menu.</td>
</tr>
</tbody>
</table>

Table 7: Model merge sequence of steps.

Once the projects have been combined into an overall model, time analysis may be run to determine the first cut schedule of activities tracked by the IPPS. Before time analysis is run, the only dates in the system are the fixed dates on which blocks or grand blocks erect to the ship. Time analysis will create three sets of dates for each activity: early, late, and scheduled. The model is constrained only by the back end erection activities.
Since there is no constraint placed on early dates, the only dates that have any meaning to the system are the late dates. All programs within the IPPS use late dates. The sequence of steps to perform time analysis is explained in the Open Plan manual. The processed yardwide model must be analyzed to see if the manning and facility utilization implied by the schedule are acceptable. Manning and facilities are investigated by creating resource utilization curves and laydown charts. Resource utilization data is generated by creating the Aggregation File for the overall model. The Aggregation File is then processed by the Data Program of the IPPS. The dBase file created by the Data Program is imported into a graphics software package and resource utilization curves are produced. Table 8 shows the sequence of steps necessary to create the resource utilization curves. For the detailed explanation of how the Build Program performs its task, see the program logic flowchart in appendix IV. A sample of the resource utilization curve created by the IPPS is shown in figure 13.

**TOTAL OUTFITTING MANNING REQUIREMENTS**

![Resource Utilization Curve](image_url)

Figure 13: Resource utilization curve developed through the IPPS for total outfitting manning requirements (resource code H123.)
Select the General Reports menu choice under the Reports heading of Open Plan’s main menu.
- Select F5 for Resource.
- Select HISTRES- Resource histogram from aggregation file.
- Enter the model name (e.g. OVERALL) of project for which you are creating the resource utilization curves.
- The system asks if you need to do aggregation: ‘Y’.
- Enter the resource availability file (e.g. OVERALL). This file is the resource availability file created in step (4) of table 6.
- Enter the aggregation file name (e.g. OVERALL).
- Enter the aggregation period length (e.g. 7 for weekly aggregation).
- Enter the aggregation start date (e.g. 12/30/91). It is best to select a Monday so that the ending date of the aggregation period will include a full work week and end on Sunday.
- Enter number of aggregation periods (e.g. 104 for 2 Years).
  You can select whatever period you feel appropriate, the time span should be at least as long as the span of the project for which you are developing schedules.
  Open Plan will create the aggregation file (OVERALL.agg in this example) containing data regarding the usage over time of each resource defined in the resource library.
  When the aggregation file is complete, the system returns asking for resource codes for reporting purposes. At this point, escape from the system by entering ‘*’.
  Enter ‘*’ again to return to Open Plan’s main menu.

Select the Integrated Plan System menu choice under the Process heading of Open Plan’s main menu.
- Enter ‘2’: Project Analysis.
- Enter ‘l’: Data Generation Program.
- Enter the aggregation file name and path as created in step (1) above (e.g. C:\OVERALL).
- Enter the curve generation file path and name as created in step (4) of table 6 (e.g. C:\CURVES).
- Enter the number of days per period, start date, and number of aggregation periods as was entered when creating the aggregation file in step (1) of this procedure.
  The data generation program will take the Aggregation File developed by Open Plan and rewrite it to the Curves Data File in a form suitable for curve generation by the graphics software.
  Escape back to Open Plan main menu.

Import the fields you are investigating (e.g. Hll if you are investigating total steel hour utilization) into a graphics software program and print out the curves. It is recommended that standard templates be developed within the graphics program for creation of resource utilization curves. The start date and time span shown on the templates should correspond to the aggregation and curves file generated in the previous steps.

Table 8: Resource utilization curve generation sequence of steps.
After resource utilization curves have been analyzed and preliminary schedule leveling has been accomplished, laydown charts are developed. These charts graphically display space utilization within a yard by showing the location of blocks as a function of time. The Laydown Generation Program uses the activity file of the Overall Production Model after processing by the project management software. The program takes the activity file and extracts the dates and locations necessary to develop laydown schedules for each production area. For a detailed explanation of how the Laydown Generation Program performs its task, see the program logic flowchart in appendix IV. The IPPS as currently developed has no automated method for assigning laydown positions to activities. The assignment is made by manually inputting the laydown locations into the “TABLE” field of the Overall Model activity file. Automation of the laydown position assignment process is an area which the system users may wish to explore. Once all activities for which laydown charts are developed have been assigned a laydown location, the laydown charts may be produced. The sequence of steps required to produce the laydown charts are shown in table 9. A sample of a laydown chart created by the IPPS is shown in figure 14.

(1) Select the Integrated Plan System menu choice under the Process heading of Open Plan’s main menu.

(2) Select 4: Production Schedule Update.

(3) Select 3: Laydown Schedules.

(4) Enter the model name with path (e.g. C:\OVERALL)’.

The program will set up required fields and re-index.

(5) The program will ask if you wish to recalculate placements. If laydown locations or schedule changes have been made since the laydown was last processed, enter ‘Y’. If no changes have been made since the last laydown processing, enter ‘N’ and the program advances to the next step without recalculating.

(6) Place an ‘X’ by all areas for which you wish laydown tables printed. Place an ‘X’ by all time periods for which you wish laydowns printed.

The program will produce the laydown charts requested.

(7) Escape back to the Open Plan main menu.

Table 9: Laydown chart generation sequence of steps.
Figure 14: Laydown charts produced by the Integrated Production Planning System showing the scheduled laydown of blocks in outfitting area 'B' for the M/V WELL PLANNED.
Once scheduling iterations have been performed and a credible schedule developed, the IPPS will produce Gantt charts for each of the blocks and grand blocks. Table 10 shows the sequence of steps necessary to create the Gantt charts. For a detailed explanation of how the Gantt Generation Program performs its task, see the program logic flowchart in appendix IV.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the Integrated Plan System menu choice under the Process heading of Open Plan’s main menu.</td>
</tr>
<tr>
<td>2</td>
<td>Select 3: Baseline Schedule Generation.</td>
</tr>
<tr>
<td>3</td>
<td>Select 2: Master Schedules (Gantt Chart Form).</td>
</tr>
<tr>
<td>4</td>
<td>Enter the model name with path of project for which you wish to generate Gantt charts (e.g. C:\OVERALL).</td>
</tr>
<tr>
<td>5</td>
<td>Enter the contract letter of the block for which you wish to view the Gantt chart.</td>
</tr>
<tr>
<td>6</td>
<td>Enter the block number for which you wish to view a Gantt chart. The Gantt chart and table of activity dates for the block requested will appear on the screen. The program asks if you wish a hard copy.</td>
</tr>
<tr>
<td>7</td>
<td>To print a hard copy of the Gantt chart enter ‘Y’.</td>
</tr>
<tr>
<td>8</td>
<td>Escape back to Open Plan main menu.</td>
</tr>
</tbody>
</table>

Table 10: Gantt chart generation sequence of steps.

When a Gantt chart for a grand block is requested, only the activities at the grand block level appear on the screen. When a hard copy of the Gantt chart is printed, both the activities for the grand block and the blocks stacking to the grand block are shown. A sample of a grand block Gantt chart produced by the IPPS is shown in figure 15.
### GRAND BLOCK 510

**BUILD STRATEGY**

<table>
<thead>
<tr>
<th>Block 010</th>
<th>Start/Completion Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Stll&quot; Fabrication</td>
<td>09/24/92 - 09/24/92</td>
</tr>
<tr>
<td>&quot;Stll&quot; Assembly</td>
<td>10/25/92 - 12/01/92</td>
</tr>
<tr>
<td>&quot;Stll&quot; Sub-Assembly</td>
<td>11/05/92 - 01/12/93</td>
</tr>
<tr>
<td>&quot;Stll&quot; Stacking</td>
<td>01/13/93 - 02/02/93</td>
</tr>
</tbody>
</table>

**Block 100**

<table>
<thead>
<tr>
<th>&quot;Stll&quot;</th>
<th>Start/Completion Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Stll&quot; Sub-Assembly</td>
<td>10/12/92 - 10/12/92</td>
</tr>
<tr>
<td>&quot;Stll&quot; Fabrication</td>
<td>10/12/92 - 12/08/92</td>
</tr>
<tr>
<td>&quot;Stll&quot; Assembly</td>
<td>11/09/92 - 01/12/93</td>
</tr>
<tr>
<td>&quot;Stll&quot; Stacking</td>
<td>12/14/92 - 01/12/93</td>
</tr>
</tbody>
</table>

**Block 510**

<table>
<thead>
<tr>
<th>&quot;O/F&quot; Pre-Blast (Up-Rght)</th>
<th>Start/Completion Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;O/F&quot; Blast &amp; Paint</td>
<td>02/03/93 - 02/03/93</td>
</tr>
<tr>
<td>&quot;O/F&quot; Post Blast</td>
<td>02/03/93 - 02/03/93</td>
</tr>
</tbody>
</table>

---

### GRAND BLOCK 510

**BUILD STRATEGY**

**Block Number: 010**

<table>
<thead>
<tr>
<th>STEEL</th>
<th>Start</th>
<th>Completion</th>
<th>Dur</th>
<th>OUTFITTING</th>
<th>Start</th>
<th>Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fab(711)</td>
<td>09/24/92</td>
<td>12/01/92</td>
<td>40</td>
<td>O/F Inv.(7251)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub-Assy(712)</td>
<td>11/05/92</td>
<td>01/12/93</td>
<td>40</td>
<td>O/F Up-R(7252)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assembly(713)</td>
<td>10/25/92</td>
<td>12/01/92</td>
<td>20</td>
<td>S/B &amp; P(7253)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stacking(715)</td>
<td>01/13/93</td>
<td>02/02/93</td>
<td>15</td>
<td>O/F Post(7254)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Block Number: 100**

<table>
<thead>
<tr>
<th>STEEL</th>
<th>Start</th>
<th>Completion</th>
<th>Dur</th>
<th>OUTFITTING</th>
<th>Start</th>
<th>Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fab(711)</td>
<td>10/12/92</td>
<td>12/08/92</td>
<td>40</td>
<td>O/F Inv.(7251)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub-Assy(712)</td>
<td>10/12/92</td>
<td>12/08/92</td>
<td>40</td>
<td>O/F Up-R(7252)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assembly(713)</td>
<td>11/09/92</td>
<td>12/08/92</td>
<td>20</td>
<td>S/B &amp; P(7253)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stacking(715)</td>
<td>12/14/92</td>
<td>01/12/93</td>
<td>15</td>
<td>O/F Post(7254)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Block Number: 510**

<table>
<thead>
<tr>
<th>STEEL</th>
<th>Start</th>
<th>Completion</th>
<th>Dur</th>
<th>OUTFITTING</th>
<th>Start</th>
<th>Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fab(711)</td>
<td></td>
<td></td>
<td></td>
<td>O/F Inv.(7251)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub-Assy(712)</td>
<td></td>
<td></td>
<td></td>
<td>O/F Up-R(7252)</td>
<td>02/03/93</td>
<td>02/09/93</td>
</tr>
<tr>
<td>Assembly(713)</td>
<td></td>
<td></td>
<td></td>
<td>S/B &amp; P(7253)</td>
<td>02/10/93</td>
<td>02/23/93</td>
</tr>
<tr>
<td>Stacking(715)</td>
<td></td>
<td></td>
<td></td>
<td>O/F Post(7254)</td>
<td>02/24/93</td>
<td>03/02/93</td>
</tr>
</tbody>
</table>

---

Figure 15: Gantt charts produced by the Integrated Production Planning System for Grand Block 510 of the MV WELL PLANNED.
Progressing and Updating the Model

A schedule is a dynamic rather than a static document. For schedules to remain current and meaningful, they must be updated based upon input from all affected parties. This input may be received at regularly scheduled meetings where production plans are updated to reflect actual and projected progress. These meetings are described in the Project Report section of this development. The IPPS model may be updated and progressed as explained in the Open Plan User's manual. Revised schedules, manning curves, and laydowns are produced from the updated model.

CUSTOMIZING THE SYSTEM FOR USE IN YOUR YARD

As stated earlier, the Integrated Production Planning System presented in this report is not intended to be a turn key system. To apply the methods and techniques of this report, a yard must establish a team to develop a planning system. The IPPS should not be viewed as a computer system, it is a production planning system that makes use of computer tools. Simply obtaining and installing the software will not give a shipyard an operable system. To be successful, a yard must expend the time and effort necessary to develop and maintain the system. Coding must be altered to reflect the yard's facilities, production strategies, organization, and work breakdown. Decisions must be made as to which resources should be tracked by the IPPS. Developing an Integrated Production Planning System is a significant task. However, once developed, the IPPS is a valuable tool that will assist shipyard personnel in making effective production decisions.
If you can track us down, we would like to help. Our number and address as of Spring ’92 is shown below. Please do not hesitate to write or call.

Rich Neumann or Dave McQuaide
MS 07
National Steel & Shipbuilding Co.
Harbor Drive & 28th St.
San Diego, Ca. 92186-5278

call (619) 544-3583 for Rich Neumann
or (619) 544-8481 for Dave McQuaide
APPENDIX

INTEGRATED PRODUCTION PLANNING SYSTEM
APPENDIX

I. LISTING OF THE FILE STRUCTURES USED BY THE IPPS
II. STRATEGY SHEET LOCATION OF BUILD DATA FILE INFORMATION
III. OBS CODE DATA FILE
IV. PROGRAM LOGIC FLOW CHARTS
   Ž MENU SYSTEM
   Ž MODEL BUILDER
   Ž RESOURCE UTILIZATION CURVE GENERATION
   Ž LAYDOWN CHART GENERATION
   Ž GANTT CHART GENERATION
V. PROGRAM CODE
VI. APPLICATION OF PC BASED FACILITIES SIMULATION
APPENDIX I

LISTING OF THE FILE STRUCTURES
USED BY THE IPPS
**BUILD. dbf - Build Data File**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK</td>
<td>Character</td>
<td>3</td>
</tr>
<tr>
<td>BLK_TYPE</td>
<td>Character</td>
<td>12</td>
</tr>
<tr>
<td>STRATEGY</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>GRAND</td>
<td>Character</td>
<td>3</td>
</tr>
<tr>
<td>ERECTION</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>LAG_E</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>ASSY_POS</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>OF1_D</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>OF2_D</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>OF3_D</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>OF4_D</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BUD_OF1</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BuD_of2</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BuD_of3</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BuD_of4</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BUD_FAB</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BUD_SUB</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BUD ASSY</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>BUD_ERECT</td>
<td>Numeric</td>
<td>6</td>
</tr>
<tr>
<td>FIRST_S</td>
<td>Character</td>
<td>3</td>
</tr>
</tbody>
</table>

**P_LANE.dbf - Process Lane Data File**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>PRED</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>TYPE</td>
<td>Character</td>
<td>2</td>
</tr>
<tr>
<td>GRAND</td>
<td>Character</td>
<td>3</td>
</tr>
<tr>
<td>LAG</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>RELTF</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>RELFF</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>PLOTJY</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLOTIX</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLOTAY</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLOTJX</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>RELCAL</td>
<td>Numeric</td>
<td>3</td>
</tr>
</tbody>
</table>

**WBS_C1.cod - Code File 1 (WBS)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>CIDESC</td>
<td>Character</td>
<td>60</td>
</tr>
</tbody>
</table>
### OBS_C2 . cod - Code File 2 (OBS)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>C2DESC</td>
<td>Character</td>
<td>60</td>
</tr>
</tbody>
</table>

### GRD C4. cod - Code File 4 (Grand Block Code)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>C4DESC</td>
<td>Character</td>
<td>60</td>
</tr>
</tbody>
</table>

### PROJECT.cal - Calender Data File

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALENDER</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>HOLIDAYFIL</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DAY1</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>DAY2</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>DAY3</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>DAY4</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>DAY5</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>DAY6</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>DAY7</td>
<td>Numeric</td>
<td>1</td>
</tr>
</tbody>
</table>

### PROJECT.hol - Holiday Data File

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOLIDAY</td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

### PROJECT.rds - Resource Library Data File

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESCODE</td>
<td>Character</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>RESDESC</td>
<td>Character</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>UNIT</td>
<td>Character</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>UNITCOST</td>
<td>Numeric</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>THRESHOLD</td>
<td>Numeric</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>RESTYPE</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>RESCLASS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SUPPRESS</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
### Model.act & ModelA.act - Activity Data File for Projects Model and ModelA

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>CAL</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>DS</td>
<td>Character</td>
<td>30</td>
</tr>
<tr>
<td>TYPE</td>
<td>Character</td>
<td>1</td>
</tr>
<tr>
<td>PROG</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>TARGS</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>TARGF</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>C1</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>C2</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>C3</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>C4</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>FF</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>TF</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>ESDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>EFDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>LSDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>LFDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>SSDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>SFDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>CRITICAL</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>LOGICODE</td>
<td>Character</td>
<td>1</td>
</tr>
<tr>
<td>COMPSTAT</td>
<td>Numeric</td>
<td>1</td>
</tr>
<tr>
<td>ACTS</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>ACTF</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>BSTART</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>BFINISH</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>BCOST</td>
<td>Numeric</td>
<td>9</td>
</tr>
<tr>
<td>ACOST</td>
<td>Numeric</td>
<td>9</td>
</tr>
<tr>
<td>PCOMP</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>ACOST_TLP</td>
<td>Numeric</td>
<td>9</td>
</tr>
<tr>
<td>PCOMP_TLP</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>RESCOST</td>
<td>Numeric</td>
<td>9</td>
</tr>
<tr>
<td>TARGSTYPE</td>
<td>Character</td>
<td>2</td>
</tr>
<tr>
<td>TARGFTYPE</td>
<td>Character</td>
<td>2</td>
</tr>
<tr>
<td>FEDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>DELAYRES</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>COMPREMDUR</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>XFDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>ACTYPE</td>
<td>Character</td>
<td>1</td>
</tr>
<tr>
<td>MAXD</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>MAXSPLITS</td>
<td>Numeric</td>
<td>2</td>
</tr>
<tr>
<td>MIN_SPLITD</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>RSCLASS</td>
<td>Character</td>
<td>1</td>
</tr>
<tr>
<td>RESLABUNIT</td>
<td>Numeric</td>
<td>9</td>
</tr>
<tr>
<td>PTF</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>SDUR</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLACEMENT</td>
<td>Character</td>
<td>2</td>
</tr>
<tr>
<td>TABLE</td>
<td>Character</td>
<td>5</td>
</tr>
<tr>
<td>SIZE</td>
<td>Character</td>
<td>5</td>
</tr>
</tbody>
</table>
MODEL. res & MODELA. res - Resource Data File for projects MODEL and MODELA

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>RESCODE</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>LEVEL</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>OFFSET</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>PERIOD</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>LEVTYPE</td>
<td>Character</td>
<td>1</td>
</tr>
</tbody>
</table>

MODEL. rel & MODELA.rel - Relationship Data File for projects MODEL and MODELA

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>PRED</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>TYPE</td>
<td>Character</td>
<td>2</td>
</tr>
<tr>
<td>LAG</td>
<td>Numeric</td>
<td>3</td>
</tr>
<tr>
<td>RELTF</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>RELFF</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>PLOTJY</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLOTIX</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLOTAY</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLOTIY</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>PLOTJX</td>
<td>Numeric</td>
<td>4</td>
</tr>
<tr>
<td>RELCAL</td>
<td>Numeric</td>
<td>3</td>
</tr>
</tbody>
</table>

PROJECT. aVl - Resource Availability Data File

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESCODE</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>LEVEL</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>FROMDATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>TODATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>RESCAL</td>
<td>Numeric</td>
<td>3</td>
</tr>
</tbody>
</table>
**Curve . dbf - Curve Generation Data File**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODATE</td>
<td>Date</td>
<td>8</td>
</tr>
<tr>
<td>WORKPDS</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H11</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H112</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H1131</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H1132</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H1133</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H113</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H114</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H1231</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H1232</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H123</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H124</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H11</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H12</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>H1</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P A</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>PEE</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P1131</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P1132</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P1133</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P113</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P1231</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P1232</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P123</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P124</td>
<td>Numeric</td>
<td>5</td>
</tr>
<tr>
<td>P12</td>
<td>Numeric</td>
<td>5</td>
</tr>
</tbody>
</table>
APPENDIX II

STRATEGY SHEET LOCATION OF BUILD DATA FILE INFORMATION
Erection "STAR" Chart showing data gathered for use by the Model Builder Program.
Integrated assembly/outfit strategies showing data gathered for use by the Model Builder Program.
GRAND BLOCK STRATEGY
LOWER HOUSE - G.B. 531

<table>
<thead>
<tr>
<th>BLK TYPE</th>
<th>BLOCK NUMBER</th>
<th>GRAND</th>
<th>BUILD STRAT.</th>
<th>ASSY</th>
<th>PRE-BLAST INVERTED</th>
<th>PRE-BLAST UP RIGHT</th>
<th>BLAST &amp; PAINT</th>
<th>POST BLAST</th>
<th>ERECT/STACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOUSE GB</td>
<td>531</td>
<td>531</td>
<td>GRAND2</td>
<td></td>
<td>40</td>
<td>2500</td>
<td>5</td>
<td>15</td>
<td>20</td>
</tr>
</tbody>
</table>

Grand block strategy sheet showing data gathered for use by the Model Builder Program.
PROCESS LANE STRATEGIES

ASSEMBLY PROCESS LANE FOR WING TANKS

BLOCK
151
152
141
142
131
132
121
122

SINGLE ASSEMBLY POSITION

KEY
♦ ERECTION
◻ ADDED FLOAT
■ POST BLAST
□ BLAST & PAINT

= PRE-BLAST (UP RIGHT)
= PRE-BLAST (INVERTED)
= ASSEMBLY

Process lane strategies showing data gathered for use by the Model Builder Program.

FINISH TO START RELATION FROM ASSY OF BLOCK 121 TO ASSY OF BLOCK 122
APPENDIX III

OBS CODE DATA FILE
<table>
<thead>
<tr>
<th>CODE FIELD 2 (C2)</th>
<th>DESCRIPTION (C2DESC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Production</td>
</tr>
<tr>
<td>11</td>
<td>Steel Area’s</td>
</tr>
<tr>
<td>12</td>
<td>Non-Steel Areas</td>
</tr>
<tr>
<td>111</td>
<td>Steel Fabrication</td>
</tr>
<tr>
<td>112</td>
<td>Steel Sub-Assembly</td>
</tr>
<tr>
<td>113</td>
<td>Steel Assembly</td>
</tr>
<tr>
<td>114</td>
<td>Steel Erection</td>
</tr>
<tr>
<td>1131</td>
<td>Steel Assembly - Flat Block</td>
</tr>
<tr>
<td>1132</td>
<td>Steel Assembly - Curved Block</td>
</tr>
<tr>
<td>1141</td>
<td>Erection - On-Board</td>
</tr>
<tr>
<td>1142</td>
<td>Erection - Grand Block Stacking</td>
</tr>
<tr>
<td>121</td>
<td>Pipe Shop</td>
</tr>
<tr>
<td>122</td>
<td>Vent Shop</td>
</tr>
<tr>
<td>123</td>
<td>Outfitting</td>
</tr>
<tr>
<td>124</td>
<td>Blast &amp; Paint</td>
</tr>
<tr>
<td>1231</td>
<td>Outfitting - Pre-Blast</td>
</tr>
<tr>
<td>1232</td>
<td>Outfitting - Post Blast</td>
</tr>
<tr>
<td>12311</td>
<td>Outfitting - Pre-Blast (Inverted)</td>
</tr>
<tr>
<td>12312</td>
<td>Outfitting - Pre-Blast (Upright)</td>
</tr>
</tbody>
</table>
APPENDIX IV

PROGRAM LOGIC FLOW CHARTS
FOR

Ž MENU SYSTEM
Ž MODEL BUILDER
Ž RESOURCE UTILIZATION CURVE GENERATION
Ž LAYDOWN CHART GENERATION
Ž GANIT CHART GENERATION
SP8_MENU.prg

LOGIC DIAGRAM

Return to Open Plan

Set Procedure to SP8_MENU

IF "Esc" Pressed

YES

NO

CALL

Read Menu Choice

CALL

Process Menu Choice

CASE:
Choice = "1"

Close Procedure

CALL

Program SP8_M1

Set Procedure to SP8_MENU

CASE:
Choice = "2"

Close Procedure

CALL

Program SP8_M2

Set Procedure to SP8_MENU

CASE:
Choice = "3"

Close Procedure

CALL

Program SP8_M3

Set Procedure to SP8_MENU

CASE:
Choice = "4"

Close Procedure

CALL

Program SP8_M4

Set Procedure to SP8_MENU
SP8_M1.prg
LOGIC DIAGRAM

Return to SP8_M1

Set Procedure to SP8_M1

IF "Esc" Pressed

YES

CALL

Procedure MENU_1

Screen Display Format

NO

Read Menu Choice

CALL

Procedure MENU_1A

Shaded Screen Display Format

Process Menu Choice

CASE: Choice = 1

CALL

Close Procedure

Program MODEL_B

CALL

Set Procedure to SP8_M1

CALL

Set Procedure to SP8_M1
MODEL_B.prg
LOGIC DIAGRAM

1. Set-Up Menu Variables
   CALL
   Procedure MODELM

2. Read Menu Variables

3. IF "Esc" Pressed
   YES
   EXIT
   NO

4. Create Database Variables from Menu Choices

5. Open & Clear All Records from the Activity, Relation and Resource Files
   RESOURCE File in Section "D"; Alias "RES"
   RELATION File in Section "C"; Alias "REL"
   ACTIVITY File in Section "B"; Alias "ACT"

6. Open and Index the Raw Data File
   DATA File in Section "A"; Alias "BUILD"

7. Select "BUILD"

8. IF End of File
   YES
   Goto 40
   NO

9. Set-Up Screen Information Format

Continued on Page 2
Continued from Page 3

Build Strategy

25

Up-Date Screen Information

CALL

Procedure ACTG2

CALL

Procedure RELC2

CALL

Procedure RESQ2

Goto 30

CASE: Strategy = "SHOP"

Up-Date Screen Information

CALL

Procedure ACTI

CALL

Procedure RELS

CALL

Procedure RESI1

30

Up-Date Screen Information

Select "BUILD"

Next Record

Return to 10
Continued from Page 7

Select "ACT"

Copy to Temp. File for Unique Data in field "C3"

Select "CODE"

Delete all Existing Records in Code File

Append Code File from Temp. File

Create Code File "C4"

YES

READ Code File Name

Does File Exist

NO

Display Screen "Error" Message

YES

Select "ACT"

Copy to Temp. File for Unique Data in field "C4"

Select "CODE"

Delete all Existing Records in Code File

Append Code File from Temp. File

RETURN TO MAIN MENU

Page:
PROCEDURE ACT1
LOGIC DIAGRAM
Procedures ACTG1, ACT2 & ACTG2 are Similar

NOTE:
The following Fields are filled in for each Activity

ID - ID Number
DS - Description
D - Duration
CAL - Calendar
C1 - Code Field #1
C2 - Code Field #2
C3 - Code Field #3
C4 - Code Field #4

Return to Calling Program
PROCEDURE REL1

LOGIC DIAGRAM

Procedures RELG1, REL2, RELG2 & REL3 are Similar

1. Update Screen Information
2. IF Grand = '
   - YES: ADD Relation Record for "Erection"
   - NO: ADD Relation Record for "Stacking"
3. ADD Relation Record for "Outfitting" (Post Blast)
4. ADD Relation Record for "Blast & Paint"
5. ADD Relation Record for "Outfitting" (Pre-Blast Upright)
6. ADD Relation Record for "Outfitting" (Pre-Blast Inverted)
7. ADD Relation Record for "Assembly"
8. ADD Relation Record for "Sub-Assembly"
9. ADD Relation Record for "Fabrication"
10. Return to Calling Program

NOTE:

The following Fields are filled in for each Relation

ID - ID Number
PRED - Predecessor ID Number
TYPE - Relation Type; "FS", "FF", "SF", "SS"
LAG - Days between Relation
PROCEDURE RES1

LOGIC DIAGRAM

Procedures RESG1, RES2 & RESG2 are Similar

NOTE:
The following Fields are filled in for each Resource

ID - *ID Number*
RESCODE - *Code Name for the Resource*
LEVEL - *Quantity of Resource*
LEVTYPE - *How to spread Resource*

Return to Calling Program
PROCEDURE DATA1
LOGIC DIAGRAM
Procedure DATA2 is similar
Continued from Page 1

Setup
Print Header Variables

Setup
Background Variables

Call → Procedure DATE_BAR

Call → Procedure GANTT_1

Return to 10
PROCEDURE DATE_BAR

LOGIC DIAGRAM

CASE:
If
"1" (January)

CASE:
All Cases are Similar to "January"
February May September
March June October
April July November
August December

Store "Date Bar" Information & Shading to Print Variables for Dates Starting in January

Return to Calling Program
Continued from Page 1

- If Block > 500
  - NO
  - YES  
    - Select "ACTIVITY"
    - Copy all Activities related to the Grand Block to a Temp File
    - Select "D"
    - Open Temp File
    - Send Printer Codes to the Printer
      - CALL Program G_PAGE_D
      - CALL Program G_PAGE_G
      - Return to Calling Program

- Select "ACTIVITY"
  - Copy all Activities related to the Block to a Temp File
  - Select "D"
  - Open Temp File
  - Send Printer Codes to the Printer
    - CALL Program G_PAGE_A
    - Return to Calling Program
PROCEDURE G_PAGE_D

LOGIC DIAGRAM

Setup Memory Variables

Print Grand Block Information

IF End of File

YES

NO

Store Block Info. to Variable

Call

Procedure G_EKKI
SP8_M4.prg
LOGIC DIAGRAM

Return to SP8-MENU.prg

Set Procedure to SP8_M4

IF "Esc" Pressed

YES

NO

CALL

Procedure MENU_1

Screen Display Format

CALL

Procedure MENU_1A

Shaded Screen Display Format

CALL

Process Menu Choice

CASE:
Choice = "1"

Display Text
Describing this Future Option

CASE:
Choice = "2"

Display Text
Describing this Future Option

CASE:
Choice = "3"

Close Procedure

CALL

Set Procedure to SP8-M4

Program SP8_LAY
continued on page 1

Set-Up Menu Variables

Set-Up Screen Text (Area & Date) Selections

Read Menu Selections

Set-Up Background Shading Variables

Set-Up Start/Comp Dates Date Bar Variables

Set-Up Date Bar Text Variables

Set-Up Placement Condition Variables

IF Shop Area Selected

YES

Store "Shop" Information to Title Variables

Update Screen Information

Store "Shop" to BODY Variable

These steps are similar for each area selected:

Table 1
Table 2
Area A
Area B

Procedure PLACEMENT

NO

Call

Return to SP6-M4.prg
PROCEDURE PRT_LAY

LOGIC DIAGRAM

Update Print Variables

Send Printer Codes to Printer

IF First Half of 1992 Selected

YES

Update Screen Information

Store Date Bar Variables to Header Variables

Store Start/Comp. Parameters to Date Variables

NO

Call Procedure HEADING

These Step are Similar for Each Time Period Selected.

Second Half 1992
First Half 1993
Second Half 1993

Call Procedure [BODY]

Return to SP8-M4.prg
PROCEDURE TABLE_1
LOGIC DIAGRAM
Procedures SHOP, TABLE_2, AREA_A & AREA_B are Similar

Store Table Position to Seek Variable
Table 1: Position 1 [01-01]

Call

Procedure DIVIDE
Program Code to Print a Horizontal Divider Line

Store Position Info. to Title Variable

Call

Procedure LINE
Program Code to Print a Background Shading

Store Print Condition 1 to Variable

Call

Procedure GANTT_in

Advance to Next Line

Store Position Info. to Title Variable

Call

Procedure LINE
Program Code to Print a Background Shading

Call

Procedure DATA_in

Advance to Next Line

These Steps are Repeated for Print Condition 2

Call

Procedure DIVIDE
Program Code to Print a Horizontal Divider Line

These Steps are Repeated for Each Position on Table 1
[01-02] [01-05] [01-08]
[01-03] [01-06] [01-09]
[01-04] [01-07] [01-10]

Call

Procedure BOTTOM
Program Code for Printing the Bottom Line of the Area Laydown Form. Also Calls Procedure LEGEND, which Prints the Legend below the Bottom Line.

Return to SP6-M4.prg
PROCEDURE GANTT_In

LOGIC DIAGRAM

Procedure DATA_In is similar

Seek for Table Position

IF End of File

YES

NO

IF TABLE = Table Var

YES

Record within Gantt Dates

NO

If Gantt Bar Start & Comp. fall within Date parameters

YES

NO

If Gantt Bar Start <= Complete

YES

NO

If Condition Matches

YES

NO

Store Display Character to Gantt Variable

Print Gantt Bar according to the Condition Detailed in Program Code

Next Record

Return to Calling Program
APPENDIX V

PROGRAM CODE
INPUT FILE: C:\OPLAN\SP8_MENU.PRG

1 * FILE NAME: SP8_MENU.PRG
2 * BY: D. McQuaide
3 * DATE: March 24, 1992
4 * DESC:
5 * CALLED BY:
6 * DATA FILES:
7 * SP8_MENU.prg

8 close all
9 set talk off
10 set status off
11 set safety off
12 set color to gr/ w/r
13 clear
14 set procedure to SP8_MENU

15 do while .T.
16   set color to gr
17     clear
18     store ' ' to Choice
19     do MENU_1
20       a 15, 38 get Choice picture '1'
21     read
22     if readkey() = 12
23         set color to w/b
24     endif
25     do MENU_1A
26       do case
27         case Choice = '1'
28           set procedure to
29           do SP8_M1
30           set procedure to SP8_MENU
31         case Choice = '2'
32           set procedure to
33           do SP8_M2
34           set procedure to SP8_MENU
35         case Choice = '3'
36           set procedure to
37           do SP8_M3
38           set procedure to SP8_MENU
39         case Choice = '4'
40           set procedure to
41           do SP8_M4
42           set procedure to SP8_MENU
43       endcase
44     enddo
45   ==RETURN
46
47 PROC MENU_1
48 set color to gr/w/b
49 a 2, 13 to 16, 54 double
50 a 2, 25 SAY "INTEGRATED PLANNING"
51 set color to w/r,w/r
52 a 3, 14 SAY "1: NEW MODEL GENERATION"
53 a 4, 14 SAY "2: PROJECT ANALYSIS"
54 a 5, 14 SAY "3: PROJEST REVIEW"
55 a 6, 14 SAY "4: PROJECT DECISION"
74  a 7, 14 SAY " "
75  a 8, 14 SAY " 3: BASELINE SCHEDULE GENERATION"
76  a 9, 14 SAY " "
77  a 10, 14 SAY " 4: PRODUCTION SCHEDULE UPDATE"
78  a 11, 14 SAY " "
79  a 12, 14 SAY " "
80  a 13, 14 SAY " "
81  a 14, 14 SAY " "
82  a 15, 14 SAY " Enter Choice : :
83  set color to gr
84  a 17, 25 say 'Press "Esc" to return'
85
86 <==return
87
88
89
90
91  "PROCEDURE MENU_1A.prg"
92  "Text for Main Menu"
93  "Shaded to allow for Overlay"
94
95
96  PROC MENU_1A
97
98  set color to /b
99  a 2, 13 to 16, 54 double
100  a 2, 25 SAY " INTEGRATED PLANNING"
101  set color to w/bg
102  a 3, 14 SAY " "
103  a 4, 14 SAY " 1: NEW MODEL GENERATION"
104  a 5, 14 SAY " "
105  a 6, 14 SAY " 2: PROJECT INTEGRATION"
106  a 7, 14 SAY " "
107  a 8, 14 SAY " 3: BASELINE SCHEDULE GENERATION"
108  a 9, 14 SAY " "
109  a 10, 14 SAY " 4: PRODUCTION SCHEDULE UPDATE"
110  a 11, 14 SAY " "
111  a 12, 14 SAY " "
112  a 13, 14 SAY " "
113  a 14, 14 SAY " "
114  a 15, 14 SAY " Enter Choice : :
115  set color to gr
116  a 17, 25 say 'Press "Esc" to return'
117
118 <==return
119
120  "Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:45 AM. dANALYST found 0 error(s), 0 warning(s), 120 lines."
INPUT FILE: C:\PLAN\SP8_M1.PRG

* FILE NAME: SP8_M1.PRG
* BY: D. McQuade
* DATE: March 24, 1992
* DESC:
* CALLED BY:
* DATA FILES:
* SP8_M1.prg

set procedure to SP8_M1

do while ..T.
  set color to gr
  a 3, 15 clear
  store ' ' to Choice1
  do MENU_1
  a 16, 40 set Choice1 picture ' !'
  read
  if readkey() = 12
    enddo
  endif
  do MENU_1A
  if Choice1 = ' !'
    set procedure to
    do MODEL_B
    close all
    set procedure to SP8_M1
  endcase
  enddo

<== return

***********************************************************************

PROCEDURE MENU_1.prg
* Text for New Model Generation Menu *
***********************************************************************

PROC MENU_1

set color to gr/r
a 3, 15 to 17, 56 double
a 3, 26 SAY " NEW MODEL GENERATION "
set color to w/r
a 4, 16 SAY " "

a 5, 16 SAY " 1: MODEL BUILDER PROGRAM "

a 6, 16 SAY " "

a 7, 16 SAY " "

a 8, 16 SAY " "

a 9, 16 SAY " "

a 10, 16 SAY " "

a 11, 16 SAY " "

a 12, 16 SAY " "

a 13, 16 SAY " "

a 14, 16 SAY " "

a 15, 16 SAY " "

a 16, 16 SAY " Enter Choice : : "

set color to gr
a 18, 25 say 'Press "Esc" to return'

<== return

***********************************************************************

PROCEDURE MENU_1A.prg
* Text for New Model Generation Menu *
* Shaded to allow for Overlay *
***********************************************************************
PROC MENU_1A

set color to /b

@ 3, 15 to 17, 56 double

@ 3, 26 SAY " NEW MODEL GENERATION "

set color to w/bg

@ 4, 16 SAY " "

@ 5, 16 SAY " 1: MODEL BUILDER PROGRAM "

@ 6, 16 SAY " "

@ 7, 16 SAY " "

@ 8, 16 SAY " "

@ 9, 16 SAY " "

@ 10, 16 SAY " "

@ 11, 16 SAY " "

@ 12, 16 SAY " "

@ 13, 16 SAY " "

@ 14, 16 SAY " "

@ 15, 16 SAY " "

@ 16, 10 SAY " Enter Choice : : "

set color to gr

@ 18, 25 say 'Press "Esc" to return'

*Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:45 AM.
dANALYST found 0 error(s), 0 warning(s), 106 lines.
INPUT FILE: C:\OPLAM\SPB_M2.PRG

set procedure to SPB_M2

**************
do while .T.
   set color to gr
   a 3, 15 clear
   store ' ' to ChoiceM2
   do MENU_1
   a 16, 40 get ChoiceM2 picture ' '!
   read
   if readkey() = 12

   ===============return
   endif

   do MENU_1A
   do case
   case ChoiceM2 = '1'
   set procedure to SPB_DATA
   close all
   set procedure to SPB_M2
   endcase
   enddo

**************

******

PROC MENU_1

set color to gr+vb
a 3, 15 to 17, 56 double
a 3, 28 SAY " PROJECT ANALYSIS "
set color to w/r
a 4, 16 SAY "
a 5, 16 SAY " 1: DATA GENERATION PROGRAM
a 6, 16 SAY " Capacity Requirements. 
'a 7, 16 SAY " * Long Range Manning
a 8, 16 SAY " Requirements.
'a 9, 16 SAY 
'a 10, 16 SAY " 
'a 11, 16 SAY " 
'a 12, 16 SAY " 
'a 13, 16 SAY " 
'a 14, 16 SAY " 
'a 15, 16 SAY " 
a 16, 16 SAY " Enter Choice : : 
set color to gr
'a 18, 25 say 'Press "Esc" to return'

**************

PROC MENU_1A

* Text for Project Analysis Menu *
* Shaded to allow for Overlay *
***************

PROC MENU_1A

set color to /b

a 3, 15 to 17, 56 double

a 3, 28 SAY " PROJECT ANALYSIS "

set color to w/bg

a 4, 16 SAY " 1: DATA GENERATION PROGRAM"

a 5, 16 SAY " * Capacity Requirements."

a 6, 16 SAY " * Long Range Manning"

a 7, 16 SAY " Requirements."

a 8, 16 SAY " Enter Choice : :

set color to gr

a 10, 25 say 'Press "Esc" to return'

99 <==return

*Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:45 AM.
dANALYST found 0 error(s), 0 warning(s), 105 lines.
set procedure to SP8_M3

---do while .T.
  set color to gr
  a 3, 15 clear
  store ' ' to ChoiceM3
  do MENU_1
    a 16, 40 set ChoiceM3 picture '!!'
    read
    if readkey() = 12
      ===set===return
  endif
  do MENU_1A
---end case

---case ChoiceM3 = '1':
set procedure to
set color to gr+/b
  a 4,17 to 19,73 double
  a 4,34 say ' MASTER SCHEDULES MENU '
set color to gr+/r
  a 5,18 say ' :
  a 6,18 say ' MASTER SCHEDULES:
  a 7,18 say ' :
set color to w/r
  a 8,18 say ' The Master Schedule Generation Program uses the
  a 9,18 say ' activity file of the Baseline Production Model (after
  a 10,18 say ' processing by the project management software). The
  a 11,18 say ' program extracts the dates necessary to generate the
  a 12,18 say ' various master schedules used throughout the yard.
  a 13,18 say ' This module of the Integrated Production Planning
  a 14,18 say ' System is dependent upon the format by which a
  a 15,18 say ' specific yard produces its Master Production Schedule;
  a 16,18 say ' Therefore, the details of this system module have
  a 17,18 say ' not been developed for this presentation.
  a 18,18 say ' :
set color to gr
  a 20,33 say ' '
  wait
  a 20,0 clear
set procedure to SP8_M3

---case ChoiceM3 = '2':
set procedure to
  do SP8_GMT
  close all
set procedure to SP8_M3

---case ChoiceM3 = '3':
set procedure to
set color to gr+/b
  a 4,17 to 19,73 double
  a 4,35 say ' SCHEDULE UPLOAD MENU '
set color to gr+/r
  a 5,18 say ' :
  a 6,18 say ' SCHEDULE UPLOAD FILE:
  a 7,18 say ' :
set color to w/r
  a 8,18 say ' The Master Schedule Upload File Generation Program
  a 9,18 say ' converts the Integrated Production Planning System
  a 10,18 say ' data to a form so that it may be uploaded into the
  a 11,18 say ' yards Cost Schedule Control System (CSCS) database.
  a 12,18 say ' This allows the CSCS to be rapidly and accurately
  a 13,18 say ' updated when a change occurs to a master schedule.
PROC MENU_1

set color to green
a 3, 15 to 17, 56 double
a 3, 26 SAY "BASE LINE GENERATION"
set color to w/r
a 4, 16 SAY "1: MASTER SCHEDULES"
a 5, 16 SAY "2: MASTER SCHEDULES"
a 6, 16 SAY "(Tabular Form)"
a 7, 16 SAY "(Gantt Chart Form)"
a 8, 16 SAY "3: SCHEDULE UPLOAD PROGRAM"
a 9, 16 SAY 

in 1 win

"Enter Choice : "
set color to green
a 18, 25 SAY 'Press "Esc" to return'
in 1 win
149 $18, 25 say 'Press "Esc" to return'
150
151
152 <=return
153
154
155
156
157
158
159
160
161 *Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:46 AM.
   dANALYST found 0 error(s), 0 warning(s), 161 lines.
do while .T.
  set color to gr
  a 3, 15 clear
  store 't' to ChoiceM4
  do MENU_1
  a 16, 40 get ChoiceM4 picture '!!'
  read
  if readkey() = 12
  ===============return
  end

  do MENU_1A
  do case
  ---case ChoiceM4 = '1'
  set procedure to
  set color to gr+b
  a 4,17 to 18,73 double
  a 4,33 say 'PRODUCTION SCHEDULE MENU'
  set color to gr+r
  a 5,18 say 'PRODUCTION SCHEDULE:
  a 6,18 say'
  a 7,18 say'
  set color to w+y
  a 8,18 say 'The Production Schedule Generation Program uses the
  a 9,18 say 'activity file of the Production Update Model (after
  a 10,18 say 'processing by the project management software). The
  a 11,18 say 'program takes the activity file and extracts the
  a 12,18 say 'dates necessary to generate the updated production
  a 13,18 say 'schedules in both graphical and tabular form. These
  a 14,18 say 'schedules show planned vs. actual and projected
  a 15,18 say 'progress. Each Shipyard should create this module to
  a 16,18 say 'meet their own needs.
  a 17,18 say'
  set color to gr
  a 19,33 say '!!'
  wait
  a 19,0 clear
  set procedure to SP8_M4

  ---case ChoiceM4 = '2'
  set procedure to
  set color to gr+b
  a 4,17 to 18,73 double
  a 4,33 say 'SHORT TERM MANPOWER MENU'
  set color to gr+r
  a 5,18 say 'SHORT TERM MANPOWER:
  a 6,18 say'
  a 7,18 say'
  set color to w+y
  a 8,18 say 'The Short Term Manning Requirement Generation Program
  a 9,18 say 'is similar to the Long Range Manning Requirement
  a 10,18 say 'Generation Program. However, the short term
  a 11,18 say 'requirements are generated from the resource
  a 12,18 say 'aggregation file of the Production Update Model.
  a 13,18 say 'The Long Range Manning Program can be modified to
  a 14,18 say 'show short term projections.
  a 15,18 say '
  a 16,18 say '
  a 17,18 say '
  set color to gr
  a 19,33 say '!!'
  wait
3 19,0 clear
set procedure to SP8_M4

---case ChoiceM4 = '3'
set procedure to
do SP8_LAY
close all
set procedure to SP8_M4
endcase

==return

*****************************************************************************

* PROCEDURE MENU_1.prg *
* Text for Production Schedule Update Menu *
*****************************************************************************

PROC MENU_1

set color to gr+/b
a 3, 15 to 17, 56 double
a 3, 23 SAY " PRODUCTION SCHEDULE UPDATE "
set color to w+/r
a 4, 16 SAY " "
a 5, 16 SAY " 1: PRODUCTION SCHEDULES "
a 6, 16 SAY " "
a 7, 16 SAY " 2: SHORT TERM MANNING "
a 8, 16 SAY " "
a 9, 16 SAY " 3: LAYDOWN SCHEDULES "
a 10, 16 SAY " "
a 11, 16 SAY " "
a 12, 16 SAY " "
a 13, 16 SAY " "
a 14, 16 SAY " "
a 15, 16 SAY " "
a 16, 16 SAY " Enter Choice : :
set color to gr
a 18, 25 say 'Press "Esc" to return'

==return

*****************************************************************************

* PROCEDURE MENU_1A.prg *
* Text for Production Schedule Update Menu *
* Shaded to allow for Overlay *
*****************************************************************************

PROC MENU_1A

set color to /b
a 3, 15 to 17, 56 double
a 3, 23 SAY " PRODUCTION SCHEDULE UPDATE "
set color to w/bg
a 4, 16 SAY " "
a 5, 16 SAY " 1: PRODUCTION SCHEDULES "
a 6, 16 SAY " "
a 7, 16 SAY " 2: SHORT TERM MANNING "
a 8, 16 SAY " "
a 9, 16 SAY " 3: LAYDOWN SCHEDULES "
a 10, 16 SAY " "
a 11, 16 SAY " "
a 12, 16 SAY " "
a 13, 16 SAY " "
a 14, 16 SAY " "
a 15, 16 SAY " "
a 16, 16 SAY " Enter Choice : :
set color to gr
a 18, 25 say 'Press "Esc" to return'
149
150 ==retutn
151
152
153
154
155
156
*Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:46 AM.
dANALYST found 0 error(s), 0 warning(s), 156 lines.
INPUT FILE: C:\OPLAN\MODEL.B.PRG

* Program: MODEL_B.prg *
* Programmed by: D.J.McGualde & R.J. Neumann *
* Developed: 07/22/91 *
* Purpose: *
* *
* *
* *
* *
* *
* *
******************************************************************************
set procedure to MODEL_B

******************************************************************************

* Input Menu *

store 't' to tContract
store space(30) to tProject
store space(30) to tBuild_DB
do MODEL_M
@7,46 get tContract picture '!!!'
do while .T.
@9,23 get tProject picture '

read
if readkey() = 12

return
endif
store 't' to tLoop
store trim(tProject)+'.act' to tACT
store trim(tProject)+'.rel' to tREL
store trim(tProject)+'.res' to tRES
set color to gr+R
if .not. file(tACT)
a 14, 18 SAY ' Activity File NOT Found! '
store 'Y' to tLoop
endif
if .not. file(tREL)
a 15, 18 SAY ' Relation File NOT Found! '
store 'Y' to tLoop
endif
if .not. file(tRES)
a 16, 18 SAY ' Resource File NOT Found! '
store 'Y' to tLoop
endif
if tLoop = 'Y'
else
exit
done
******************************************************************************

set color to w/R
a 14, 18 SAY ' 
a 15, 18 SAY ' 
a 16, 18 SAY ' 
do while .T.
a 12,23 get tBuild_DB picture '!!!'
read
if readkey() = 12

return
endif
store trim(tBuild_DB)+'.dbf' to tBUILD
set color to gr+R
if .not. file(tBUILD)
a 14, 18 SAY ' Build Data File NOT Found! '
done
else
exit
done
done

set color to w/R
a 14, 18 SAY ' 

reading name of model and verifies all files exist

reading and verifying the BUILD data file
* Set-Up Data Bases *
set color to gr+/r
@ 14, 25 SAY ***
@ 14, 49 SAY ***
set color to gr+/r
@ 14, 18 SAY ' Setting Up Data Bases '.

select D
use &tRES alias RES
zap

select C
use &tREL alias REL
zap

select B
use &tACT alias ACT
zap

select A
use &tBUILD index GRD_BLK alias BUILD
reindex

set color to gr+/r
@ 14, 18 SAY '.

********************************************************************

* Start Model Building Process *
go top
do while .not. eof()
set color to gr+/r
@ 14, 18 SAY ' Add Block '.
@ 14, 30 SAY BUILD->Block
@ 15, 18 SAY ' Activity Records '.
@ 16, 18 SAY ' Relation Records '.
@ 17, 18 SAY ' Resource Records '.
********************************************************************
do case
case BUILD->STRATEGY = 'STD'
set color to gr+/r
@ 14, 35 SAY 'Standard Strategy '.
@ 14, 30 SAY BUILD->Block
"Adding ACTIVITY Records"
do ACTI
"Adding RELATION Records"
do REL1
"Adding RESOURCE Records"
do REST1

case BUILD->STRATEGY = 'GRAND1'
*** Grand Blocking (Post Blast) ***
set color to gr+/r
@ 14, 35 SAY 'Grand Block #1 Strat. '.

select BUILD
store recno() to tRecord
store BUILD->GRAND to tGrand

do while BUILD->GRAND = tGrand .and. BUILD->BLOCK < '500'
set color to gr+/r
@ 14, 30 SAY BUILD->Block
"Adding ACTIVITY Records"
do ACTI
"Adding RELATION Records"
do REL1
"Adding RESOURCE Records"
do REST1

select BUILD
skip
@ 15, 30 SAY '
@ 16, 30 SAY '
@ 17, 30 SAY '

endo
MODEL_B.PRQ  Formatted by dMAYSTAL V7.3a March 24, 1992 9:46 AM  Page 4

224  a 7, 33 say trim(tProject)
225  store ' ' to tp_Lanes
226  set color to gr+/r
227  a 8, 18 SAY ' Has successfully been created.'
228  a 9, 18 SAY ' Lane relationship via a File.'
229  a 10, 18 SAY ' (Y/N) : '
230  set color to w+/
231  a 11, 18 SAY ' Do you want to add any Process '
232  a 12, 18 SAY ' Lane relationship via a File.'
233  do while .T.
234     a 13, 18 SAY ' Enter Process Lane's File Name W/Path '
235     a 14, 23 SAY 'Lane relationship via a File.'
236     a 15, 18 SAY ' '
237     a 16, 18 SAY ' '
238     a 17, 18 SAY ' '
239     get tp_Lanes picture '!!'
240     read
241     if readkey() = 12
242     ####=.return
243     endif
244
245     if tp_Lanes = 'Y'
246        set color to w+/
247        a 9, 18 SAY ' '
248        a 11, 18 SAY ' '
249        a 12, 18 SAY ' '
250        a 13, 18 SAY ' '
251        a 14, 23 SAY 'Lane relationship via a File.'
252        do while .T.
253           a 15, 18 SAY ' '
254           a 16, 18 SAY ' Can Not Find File '
255        endwhile
257     endif
259  store trim(tp_Lane)*'.dbf' to dbp_LANE
260  if file(dbp_LANE)
261      set color to w+/
262      a 16, 18 SAY ' Adding Process Lane Relations '
263      append from dbp_LANE'
264      a 16, 18 SAY ' '
265  endif
266  else
268    a 16, 18 SAY ' Can Not Find File '
270  endif
272  store ' ' to tCodes
273  set color to w+/
276  a 7, 18 SAY ' '
277  a 8, 18 SAY ' Do you to Create New Code Files '
278  a 9, 18 SAY ' [e/ o] : '
279  a 10, 18 SAY ' '
280  a 11, 18 SAY ' '
281  a 12, 18 SAY ' '
282  a 13, 18 SAY ' '
283  a 14, 18 SAY ' '
284  set color to gr+/r
285  a 9, 31 SAY 'Y'
286  a 9, 35 SAY ' '  
287  a 9, 40 SAY ' '  
288  read
289
290  if tCodes = 'Y'
291    set color to w+/
292    a 12, 18 SAY ' Setting Up Indexes '
293    select ACT
294    a 13, 18 SAY ' C1 for Code Field #1 (WBS) '
295          index on C1 to C1
296    a 13, 18 SAY ' C4 for Code Field #4 (Grand Blk Code) '
297          index on C4 to C4
298    endif

displays that project has been created — — and asks if you wish to add process lanes to the model

gets PROCESS LANE Data File and creates process lane relationships

asks if you wish to add code fields to the model and sets up data files
```
MODEL_B.PRG  Formatted by dANALYST V7.3a March 24, 1992 9:46 AM  Page 3

149  set color to gr+r
150  @ 14, 30 SAY BUILD->Block
151  * Adding ACTIVITY Records *
152  do ACT1
153  * Adding RELATION Records *
154  do REL1
155  * Adding RESOURCE Records *
156  do RES1
157  
158  case BUILD->STRATEGY = 'GRAND2' .or. BUILD->STRATEGY = 'GRAND3' .or. BUILD->STRATEGY = 'GRAND4'
159  *** Grand Blocking (ON - ASSY) ***
160  set color to gr+r
161  @ 14, 35 SAY 'Grand Block #2 Strat. '
162  @ 14, 30 SAY BUILD->Block
163  
164  select BUILD
165  
166  store record() to tRecord
167  store BUILD->GRAND to tGrand
168  
169  do while BUILD->GRAND = tGrand .and. BUILD->BLOCK < '500'
170  set color to gr+r
171  @ 14, 30 SAY BUILD->Block
172  * Adding ACTIVITY Records *
173  do ACT2
174  * Adding RELATION Records *
175  do REL2
176  * Adding RESOURCE Records *
177  do RES2
178  
179  select BUILD
180  skip
181  @ 15, 30 SAY ' '
182  @ 16, 30 SAY ' '
183  @ 17, 30 SAY ' '
184  
185  enddo
186  set color to gr+r
187  @ 14, 30 SAY BUILD->Block
188  * Adding ACTIVITY Records *
189  store ' ' to tStack_of
190  do ACTG2
191  * Adding RELATION Records *
192  do RELG2
193  * Adding RESOURCE Records *
194  do RESG2
195  
196  case BUILD->STRATEGY = 'SHOP'
197  
198  set color to gr+r
199  @ 14, 35 SAY 'SHOP Strategy'
200  @ 14, 30 SAY BUILD->Block
201  * Adding ACTIVITY Records *
202  do ACT1
203  * Adding RELATION Records *
204  do REL3
205  * Adding RESOURCE Records *
206  do RES1
207  
208  set color to gr+r
209  @ 14, 35 SAY 'Shop Block Strategy '
210  @ 14, 30 SAY BUILD->Block
211  
212  endcase
213  
214  enddo
215  
216  store space(30) to tP_Lane
217  set color to w/r
218  @ 5, 18 SAY ' '  
219  @ 6, 18 SAY ' '  
220  @ 7, 18 SAY ' '  
221  
222  set color to gr+r
223```

Creating Activity, Resource, and Relationship files according to Block Build Strategy.
@ 13, 18 say 'store[y] to tC1,tC4
store space(30) to FC ode
@ 12, 18 say 'Varify Re-creating Code #1:
@ 12, 50 get tC1 picture '@
read
if tC1 = 'Y'
store[y] to tContinue
@ 15, 18 say 'Enter Code File W/Path
   do while .I.
      @ 14, 24 get fCode picture '!!!!!!!!!!!!!!!!!!!!!!!
      read
      if readkey() = 12
         store[Y] to tContinue
         exit
      endif
      store trim(fCode)+.cod' to tCode_file
      if file(tCode_file)
         @ 12, 18 say 'Can Not Find File
      endif
      @ 15, 18 say Can Not Find File
      loop
      endif
   enddo
   if tContinue = 'Y'
      set color to w/r, /w
      @ 12, 18 say 'Creating New Code 1 File
      @ 13, 24 say tCode_file
      select ACT
      set index to C1
      select E
      use &tCode_file alias Code
      zap
      append blank
      replace C1 with '1'
      replace CIDESC with 'Production'
      append blank
      replace C1 with '1'+tcontract
      replace CIDESC with 'Contract ''1'+tcontract''
      select ACT
      go top
      do while .not. eof()
         store substr(C1,3,4) to tBlock
         @ 15, 18 say 'For Block
            @ 15, 52 say tBlock
         select CODE
         append blank
         replace C1 with '1'+tcontract+tBlock
         replace CIDESC with 'Contract ''1'+tcontract'' Block '+tBlock'
         select ACT
         do while substr(C1,3,4) = tBlock .and. .not. eof()
            if substr(C1,7,1) = '1'
               @ 15, 37 say 'Steel
               select CODE
               append blank
               replace C1 with '1'+tcontract+tBlock+''
               replace CIDESC with 'Contract ''1'+tcontract'' Block '+tBlock' Steel
               select ACT
               do while substr(C1,7,1) = '1'
                  case C1 = '1'+tcontract+tBlock+''1''
                     select CODE
                     append blank
                     replace C1 with '1'+tcontract+tBlock+''1''
                     replace CIDESC with 'Contract ''1'+tcontract'' Block '+tBlock' Steel Fab.
                     case C1 = '1'+tcontract+tBlock+''12''
                        select CODE
                        append blank
                        replace C1 with '11'+tcontract+tBlock+''12''
                        replace CIDESC with 'Contract ''11'+tcontract'' Block '+tBlock' Steel Sub-Assy
                     case C1 = '11'+tcontract+tBlock+''12''
                        select CODE
                        append blank
                        replace C1 with '111'+tcontract+tBlock+''12''
                        replace CIDESC with 'Contract ''111'+tcontract'' Block '+tBlock' Steel Sub-Assy
```plaintext
374 case Cl = '1'+tContract+tBlock'+131'
375 select CODE
376 append blank
377 replace CI with '1'+tContract+tBlock'+131'
378 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Assembly'
379 append blank
380 replace CI with '1'+tContract+tBlock'+131'
381 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Assembly ''Flat Block''
382 case Cl = '1'+tContract+tBlock'+132'
383 select CODE
384 append blank
385 replace CI with '1'+tContract+tBlock'+132'
386 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Assembly'
387 append blank
388 replace CI with '1'+tContract+tBlock'+132'
389 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Assembly ''Curved Bloc
390 case Cl = '1'+tContract+tBlock'+133'
391 select CODE
392 append blank
393 replace CI with '1'+tContract+tBlock'+133'
394 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Assembly'
395 append blank
396 replace CI with '1'+tContract+tBlock'+133'
397 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Assembly ''Tilt Jig''
398 case Cl = '1'+tContract+tBlock'+141'
399 select CODE
400 append blank
401 replace CI with '1'+tContract+tBlock'+141'
402 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Erection'
403 append blank
404 replace CI with '1'+tContract+tBlock'+141'
405 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Erection ''On-Boar
406 case Cl = '1'+tContract+tBlock'+142'
407 select CODE
408 append blank
409 replace CI with '1'+tContract+tBlock'+142'
410 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Erection'
411 append blank
412 replace CI with '1'+tContract+tBlock'+142'
413 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Steel Erection ''Grand B
414 endcase
415 select ACT
416 skip
417 endif
418 if substr(CI,7,1) = '2'
419 @15, 37 say 'Non-Steel '
420 select CODE
421 append Blank
422 replace CI with '11'+tContract+tBlock'+2'
423 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Non-Steel'
424 select ACT
425 do while substr(CI,7,1) = '2'
426 do case
427 case CI = '11'+tContract+tBlock'+21'
428 select CODE
429 append blank
430 replace CI with '11'+tContract+tBlock'+21'
431 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Pipe Shop'
432 case CI = '11'+tContract+tBlock'+22'
433 select CODE
434 append blank
435 replace CI with '11'+tContract+tBlock'+22'
436 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Vent Shop'
437 case CI = '11'+tContract+tBlock'+2311'
438 select CODE
439 append blank
440 replace CI with '11'+tContract+tBlock'+2311'
441 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ On-Block O/F'
442 append blank
443 replace CI with '11'+tContract+tBlock'+2311'
444 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Pre-Blast O/F'
445 append blank
446 replace CI with '11'+tContract+tBlock'+2311'
447 replace CIDESC with 'Contract ''+tContract++'' Block '+tBlock++ Pre-Blast O/F ''Inverted''
```
449  case C1 = '1'+tContract+tBlock+2312:
450      select CODE
451      append blank
452      replace C1 with '1'+tContract+tBlock+2312
453      replace CIDESC with 'Contract'+'tContract'+'Block'+'tBlock' On-Block 0/F
454      append blank
455      replace C1 with '1'+tContract+tBlock+2311
456      replace CIDESC with 'Contract'+'tContract'+'Block'+'tBlock' Pre-Blast O/F
457      append blank
458      replace C1 with '1'+tContract+tBlock+2312:
459      replace CIDESC with 'Contract'+'tContract'+'Block'+'tBlock' Pre-Blast O/F "Up-Right"
460  case C1 = '1'+tContract+tBlock+2321:
461      select CODE
462      append blank
463      replace C1 with '1'+tContract+tBlock+2312
464      replace CIDESC with 'Contract'+'tContract'+'Block'+'tBlock' On-Block 0/F
465      append blank
466      replace C1 with '1'+tContract+tBlock+2321
467      replace CIDESC with 'Contract'+'tContract'+'Block'+'tBlock' Post-Blast O/F
468  case C1 = '1'+tContract+tBlock+24:
469      select CODE
470      append blank
471      replace C1 with '1'+tContract+tBlock+24
472      replace CIDESC with 'Contract'+'tContract'+'Block'+'tBlock' Blast & Paint
473      endcase
474      select ACT
475      skip
476      enddo
477      endif
478      a 15, 37 say '  
479      enddo
480      enddo
481      a 12, 18 say '  
482      a 13, 18 say '  
483      a 15, 18 say '  
484      endif
485
486 ******************************************************************************
487  a 12, 18 say ' Verify Re-creating Code #4 :  
488  a 12, 50 get tc4 picture '  
489  if tc4 = 'Y'
490      store space(30) to fCode
491      store 'Y' to tContinue
492      a 13, 18 say ' Enter Code File w/Path  
493      do while .T.
494      a 14, 24 get fCode picture '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
495      read
496      if readkey() = 12
497      store 'N' to tContinue
498      ==============exit
499      endif
500      store trim(fCode)\.COD to tCode_File
501      if file(tCode_File)
502         a 12, 18 say '  
503         a 13, 18 say '  
504         a 14, 18 say '  
505         a 15, 18 say '  
506      ==============exit
507      else
508         a 15, 18 say ' Can Not Find file  
509      ===========loop
510      endif
511      enddo
512      if tContinue = 'Y'
513      set color to w/r, /w
514      a 12, 18 say ' Creating temp File from Activity  
515      select ACT
516      set index to C4
517      copy to tempcode field C4
518      select E
519      a 12, 18 say ' Creating New Code 4 File  
520      a 13, 24 say tCode_File
521      use &tCode_File alias Code
522      zap
523      append from tempcode
replace all CDESC with 'Activity Related to Grand Block' +C4

@ 12, 10 say ' :
@ 13, 18 say ' :

endif

set unique off
set color to w/r, w/r

endif

*****************************************************************

ACT1 ------ Creates Activity File Data for Blocks

*****************************************************************

PROC ACT1

* Adding ACTIVITY Records *
set color to w**/r
@ 15, 30 SAY ' ----->
set color to w/r
select ACT
* Erection *
append blank

if BUILD->GRAND = ' ' then
replace ID with tContract+7160'+BUILD->BLOCK
replace DS with 'Erection of Block ' +BUILD->BLOCK
replace D with 20
replace TARGS with BUILD->ERECTION
replace TARGTYPE with 'ON'
replace CAL with 1
replace C1 with '1' +tContract+0'+BUILD->BLOCK+141'
replace C2 with '1' +141'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
else
replace ID with tContract+7150'+BUILD->BLOCK
replace DS with 'Stack of ' +BUILD->BLOCK+ ' to ' +BUILD->GRAND
replace D with 10
replace CAL with 1
replace C1 with '1' +tContract+0'+BUILD->BLOCK+142'
replace C2 with '1' +142'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
endif

* Post Blast O/F *
append blank
replace ID with tContract+7254'+BUILD->BLOCK
replace DS with 'Post Blast O/F of Block ' +BUILD->BLOCK
replace D with BUILD->OF4_D
replace CAL with 1
replace C1 with '1' +tContract+0'+BUILD->BLOCK+232'
replace C2 with '1' +232'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
* Blast & Painting *
append blank
replace ID with tContract+7253'+BUILD->BLOCK
replace DS with 'Blast & Painting of Block ' +BUILD->BLOCK
replace D with BUILD->OF3_D
replace CAL with 1
replace C1 with '1' +tContract+0'+BUILD->BLOCK+124'
replace C2 with '1' +124'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
* Pre Blast (U-R) O/F *
append blank
replace ID with tContract+7252'+BUILD->BLOCK
replace DS with 'Pre Blast (U-R) O/F of Block ' +BUILD->BLOCK
599 replace D with BUILD->OF2_D
600 replace CAL with 1
601 replace C1 with '11'tContract++10'BUILD->BLOCk+'2312'
602 replace C2 with '12312'
603 replace C3 with BUILD->BLK_TYPE
604 replace C4 with BUILD->GRAND
605 * Pre Blast (INV) O/F *
606 append blank
607 replace ID with tContract++7251'BUILD->BLOCk
608 replace DS with 'Pre Blast (INV) O/F of Block ++BUILD->BLOCk
609 replace D with BUILD->OF1_D
610 replace CAL with 1
611 replace C1 with '11'tContract++10'BUILD->BLOCk+'2311'
612 replace C2 with '12311'
613 replace C3 with BUILD->BLK_TYPE
614 replace C4 with BUILD->GRAND
615 * Assembly *
616 if BUILD->STRATEGY = 'STD' .or. BUILD->STRATEGY = 'GRAND1'
617 append blank
618 replace ID with tContract++7130'BUILD->BLOCk
619 replace DS with 'Assembly of Block ++BUILD->BLOCk
620 replace D with BUILD->ASSY_D
621 replace CAL with 1
622 do case
623 case BUILD->ASSY_PDS = 'F'
624 replace C1 with '11'tContract++10'BUILD->BLOCk+'1311'
625 replace C2 with '11311'
626 case BUILD->ASSY_PDS = 'C'
627 replace C1 with '11'tContract++10'BUILD->BLOCk+'1321'
628 replace C2 with '11121'
629 case BUILD->ASSY_PDS = 'T'
630 replace C1 with '11'tContract++10'BUILD->BLOCk+'1331'
631 replace C2 with '11131'
632 endcase
633 replace C3 with BUILD->BLK_TYPE
634 replace C4 with BUILD->GRAND
635 endif
636 * Sub-Assembly *
637 replace D with BUILD->OF1_D
638 replace CAL with 1
639 replace C1 with '11'tContract++10'BUILD->BLOCk+'1221'
640 replace C2 with '11121'
641 replace C3 with BUILD->BLK_TYPE
642 replace C4 with BUILD->GRAND
643 * Fabrication *
644 append blank
645 replace ID with tContract++7120'BUILD->BLOCk
646 replace DS with 'Fabrication of Block ++BUILD->BLOCk
647 replace D with BUILD->ASSY_D+20
648 replace CAL with 1
649 replace C1 with '11'tContract++10'BUILD->BLOCk+'1111'
650 replace C2 with '11111'
651 replace C3 with BUILD->BLK_TYPE
652 replace C4 with BUILD->GRAND
653 <=retum
654
655*****************************************************************************
656 * ACTG1 --> Creates Activity File Data for Grand Blocks *
657 * * *
658 * * *
659*****************************************************************************
660 PROC ACTG1
661 * Adding ACTIVITY Records *
662 set color to w#r
663 a 15, 30 SAY '----+
664 set color to w#r
665 select ACT
666 * Erection *
append blank

replace ID with 'tContract='7160'1+BUILD->BLOCK
replace DS with 'Erection of Grand Block '1+BUILD->BLOCK
replace D with 20
replace TARGS with BUILD->ERECTION
replace TARGETTYPE with 'ON'
replace CAL with 1
replace C1 with '1'+tContract='01+BUILD->BLOCK+'141'
replace C2 with '1141'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
* Post Blast O/F *
append blank

replace ID with 'tContract='7254'1+BUILD->BLOCK
replace DS with 'Post Blast O/F of GRD BLK '1+BUILD->BLOCK
replace D with BUILD->OF4_D
replace CAL with 1
replace C1 with '1'+tContract='01+BUILD->BLOCK+'232'
replace C2 with '1232'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND

<==return

******************************************************************************
* ACTG2 ---> Creates Activity File Data for Grand Blocks *
******************************************************************************

PROC ACTG2

* Adding ACTIVITY Records *
set color to "w*/r
@ 15, 30 SAY '===>'
set color to "w*/r
select ACT
* Erection *
append blank

replace ID with 'tContract='7160'1+BUILD->BLOCK
replace DS with 'Erection of Grand Block '1+BUILD->BLOCK
replace D with 20
replace TARGS with BUILD->ERECTION
replace TARGETTYPE with 'ON'
replace CAL with 1
replace C1 with '1'+tContract='01+BUILD->BLOCK+'141'
replace C2 with '1141'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
* Post Blast O/F *
append blank

replace ID with 'tContract='7254'1+BUILD->BLOCK
replace DS with 'Post Blast O/F of GRD BLK '1+BUILD->BLOCK
replace D with BUILD->OF4_D
replace CAL with 1
replace C1 with '1'+tContract='01+BUILD->BLOCK+'232'
replace C2 with '1232'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
* Blast & Painting *
append blank

replace ID with 'tContract='7253'1+BUILD->BLOCK
replace DS with 'Blast & Painting of GRD BLK '1+BUILD->BLOCK
replace D with BUILD->OF3_D
replace CAL with 1
replace C1 with '1'+tContract='01+BUILD->BLOCK+'124'
replace C2 with '124'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
* Pre Blast (U-R) O/F *
append blank

replace ID with 'tContract='7252'1+BUILD->BLOCK
replace DS with 'Pre Blast (U-R) O/F of GRD BLK '1+BUILD->BLOCK
replace D with BUILD->OF2_D
replace CAL with 1
replace C1 with '11+Contract+0'+BUILD->BLOCK+12312'
replace C2 with '12312'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND

if BUILD->STRATEGY = 'GRAND2' or BUILD->STRATEGY = 'GRAND4'
  append blank
  replace ID with tContract+72511+BUILD->BLOCK
  replace DS with 'Pre Blast (INV) 0/F of GRD BLK '+BUILD->BLOCK
  replace D with BUILD->OFF_D
  replace CAL with 1
  replace C1 with '11+tContract+0'+BUILD->BLOCK+12311'
  replace C2 with '12311'
endif

/* Assembly */

if BUILD->STRATEGY = 'GRAND2'
  append blank
  replace ID with tContract+71310+BUILD->BLOCK
  replace DS with 'Assembly of GRD BLK '+BUILD->BLOCK
  replace D with BUILD->ASSY_D
  replace CAL with 1
  replace C1 with '11+tContract+0'+BUILD->BLOCK+1131'
  replace C2 with '1131'
  replace C3 with BUILD->BLK_TYPE
  replace C4 with BUILD->GRAND
endif

<<<return

******************************************************************************

* ACT2 ----- Creates Activity File Data for Blocks *

******************************************************************************

PROC ACT2

* Adding ACTIVITY Records *
set color to w(fr
a 15, 30 SAY '-----'
set color to w/r
select ACT
* Erection *
append blank
replace ID with tContract+71510+BUILD->BLOCK
replace DS with 'Stecking of Blk '+BUILD->BLOCK' to '+BUILD->GRAND
do case
  case BUILD->STRATEGY = 'GRAND2'
    replace D with 1
  case BUILD->STRATEGY = 'GRAND3'
    replace D with 15
  case BUILD->STRATEGY = 'GRAND4'
    replace D with 5
  endcase
replace CAL with 1
replace C1 with '11+tContract+0'+BUILD->BLOCK+1142'
replace C2 with '1142'
replace C3 with BUILD->BLK_TYPE
replace C4 with BUILD->GRAND
  * Assembly *
  * append blank
  replace ID with tContract+71310+BUILD->BLOCK
  replace DS with 'Assembly of Block '+BUILD->BLOCK
  replace D with BUILD->ASSY_D
  replace CAL with 1
  replace C1 with '11+tContract+0'+BUILD->BLOCK+1131'
  replace C2 with '1131'
  replace C3 with BUILD->BLK_TYPE
  replace C4 with BUILD->GRAND
  * Sub-Assembly *
append blank
824  replace ID with tContract+'7120'+BUILD->BLOCK
825  replace DS with 'Sub-Assembly of Block '+BUILD->BLOCK
826  replace D with BUILD->ASSY_D+20
827  replace CAL with 1
828  replace C1 with '+11'+tContract+'04'+BUILD->BLOCK+'12'
829  replace C2 with '+112'
830  replace C3 with BUILD->BLK_TYPE
831  replace C4 with BUILD->GRAND
832  * Fabrication *
833  append blank
834  replace ID with tContract+'7110'+BUILD->BLOCK
835  replace DS with 'Fabrication of Block '+BUILD->BLOCK
836  replace D with BUILD->ASSY_D+20
837  replace CAL with 1
838  replace C1 with '+11'+tContract+'04'+BUILD->BLOCK+'11'
839  replace C2 with '+111'
840  replace C3 with BUILD->BLK_TYPE
841  replace C4 with BUILD->GRAND
842
843  c===>return
844
845
846  *******************************************************
847  * REL1 ----> Creates Relationship Data File for Blocks    *
848  *                                                         *
849  *******************************************************
850
851  PROC REL1
852
853  * Adding RELATION Records *
854  set color to w/r
855  a 16, 30 SAY '------>'
856  set color to w/r
857  a 15, 30 SAY '------>'
858  select REL
859
860  * Erection *
861  if BUILD->GRAND = '1'
862        append blank
863        replace ID with tContract+'7160'+BUILD->BLOCK
864        replace PRED with tContract+'7254'+BUILD->BLOCK
865        replace TYPE with 'FS'
866        replace LAG with 5
867     else
868        append blank
869        replace ID with tContract+'7150'+BUILD->BLOCK
870        replace PRED with tContract+'7254'+BUILD->BLOCK
871        replace TYPE with 'FS'
872        replace LAG with 0
873     endif
874  * Post Blast O/F *
875  append blank
876  replace ID with tContract+'7254'+BUILD->BLOCK
877  replace PRED with tContract+'7253'+BUILD->BLOCK
878  replace TYPE with 'FS'
879  replace LAG with 0
880  * Blast & Paint *
881  append blank
882  replace ID with tContract+'7253'+BUILD->BLOCK
883  replace PRED with tContract+'7252'+BUILD->BLOCK
884  replace TYPE with 'FS'
885  replace LAG with 0
886  * Pre-Blast (U-R) O/F *
887  append blank
888  replace ID with tContract+'7252'+BUILD->BLOCK
889  replace PRED with tContract+'7251'+BUILD->BLOCK
890  replace TYPE with 'FS'
891  replace LAG with 0
892  * Pre-Blast (INV) O/F *
893  append blank
894  replace ID with tContract+'7251'+BUILD->BLOCK
895  replace PRED with tContract+'7130'+BUILD->BLOCK
896  replace TYPE with 'FS'
897  replace LAG with 5
898  * Assembly for Sub-Assembly *
append blank
replace ID with tContract+ '7130'+BUILD->BLOCK
replace PRED with tContract+ '7120'+BUILD->BLOCK
replace TYPE with 'SS'
replace LAG with 20
* Assembly for Fabrication *
append blank
replace ID with tContract+ '7130'+BUILD->BLOCK
replace PRED with tContract+ '7110'+BUILD->BLOCK
replace TYPE with 'SS'
replace LAG with 20
<=="return

********************************************************************
* RELGI --> Creates Relationship Data File for
* Grand Blocks
*
********************************************************************
PROC RELGI

* Adding RELATION Records *
set color to w+/-r
8 16, 30 SAY '---->
set color to w/r
a 15, 30 SAY '---->
select REL

* Erection *
append blank
replace ID with tContract+ '7160'+BUILD->BLOCK
replace PRED with tContract+ '7254'+BUILD->BLOCK
replace TYPE with 'PS'
replace LAG with 5

* Stacking *
select BUILD
store '7254' to tStack2
goto tRecord
do while BUILD->GRAND = tGrand .and. BUILD->BLOCK < '500'
  if .not. BUILD->FIRST_S = ' ' 
    store BUILD->FIRST_S to tfirst_s
    <=="exit
  endif
endo
goto tRecord
do while BUILD->GRAND = tGrand .and. BUILD->BLOCK < '500'
  select REL
    if BUILD->FIRST_S = ' ' 
      replace ID with tContract+ '7150'+tFirst_s
    else 
      replace ID with tContract+tStack2+BUILD->GRAND
    endif
    replace PRED with tContract+ '7150'+BUILD->BLOCK
    replace TYPE with 'SS'
    replace LAG with BUILD->LAG_E
  select BUILD
  skip
endo
<=="return

********************************************************************
* RELG2 --> Creates Relationship Data File for
* Grand Blocks
* 
********************************************************************
PROC RELG2

* Adding RELATION Records *
set color to w#f/r
  a 16, 30 SAY "---->
set color to w#f/r
  a 15, 30 SAY "---->
selct REL

* Erection *
append blank
replace ID with tContract+'7160'+BUILD->BLOCK
replace PRED with tContract+'7254'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 5

* Post Blast O/F *
append blank
replace ID with tContract+'7254'+BUILD->BLOCK
replace PRED with tContract+'7253'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 0

* Blast & Paint *
append blank
replace ID with tContract+'7253'+BUILD->BLOCK
replace PRED with tContract+'7252'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 0

* Pre-Blast (UB) O/F *
if BUILD->STRATEGY = 'GRAND2' or BUILD->STRATEGY = 'GRAND4'
append blank
replace ID with tContract+'7252'+BUILD->BLOCK
replace PRED with tContract+'7251'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 0
endif

* Pre-Blast (INV) O/F *
if BUILD->STRATEGY = 'GRAND2'
append blank
replace ID with tContract+'7251'+BUILD->BLOCK
replace PRED with tContract+'7130'+BUILD->BLOCK
replace TYPE with 'SS'
replace LAG with 0
endif

* Stacking *
select BUILD
do case
case STRATEGY = 'GRAND2'
  store '7130' to tStack2
case STRATEGY = 'GRAND3'
  store '7252' to tStack2
case STRATEGY = 'GRAND4'
  store '7251' to tStack2
endcase
goto tRecord

-do while BUILD->GRAND = tGrand .and. BUILD->BLOCK < '500'
  if .not. BUILD->FIRST_S = '1'
    store BUILD->FIRST_S to tFirst_s
  endif
endo
goto tRecord

-do while BUILD->GRAND = tGrand .and. BUILD->BLOCK < '500'
select REL
  append blank
  if BUILD->FIRST_S = '1'
    replace ID with tContract+'7150'+tFirst_s
  else
    replace ID with tContract+tStack2+BUILD->GRAND
  endif
  replace PRED with tContract+'7150'+BUILD->BLOCK
  if BUILD->STRATEGY = 'GRAND3'
    replace TYPE with 'FS'
  else
    replace TYPE with 'SS'
  endif
endo
PROC REL2

* Adding RELATION Records *
set color to w+g/r
a 16, 30 SAY '-----'
set color to w/r
a 15, 30 SAY '-----'
select REL
* Erection *
append blank
replace ID with tContract^'7150'+BUILD->BLOCK
replace PRED with tContract^'7130'+BUILD->BLOCK
replace TYPE with 'FS'
* replace LAG with U
* Assembly for Sub-Assembly *
append blank
replace ID with tContract^'7130'+BUILD->BLOCK
replace PRED with tContract^'7120'+BUILD->BLOCK
replace TYPE with 'SS'
* replace LAG with 20
* Assembly for Fabrication *
append blank
replace ID with tContract^'7130'+BUILD->BLOCK
replace PRED with tContract^'7110'+BUILD->BLOCK
replace TYPE with 'SS'
replace LAG with 20

PROC REL3

* Adding RELATION Records *
set color to w+g/r
a 16, 30 SAY '-----'
set color to w/r
a 15, 30 SAY '-----'
select REL
* Erection *
if BUILD->GRAND = '
append blank
replace ID with tContract^'7160'+BUILD->BLOCK
replace PRED with tContract^'7254'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 5

else
append blank
replace ID with tContract^'7150'+BUILD->BLOCK
replace PRED with tContract^'7254'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 0

* Post Blast O/F *
append blank
replace ID with tContract+'7254'+BUILD->BLOCK
replace PRED with tContract+'7253'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 0
* Blast & Paint *
append blank
replace ID with tContract+'7253'+BUILD->BLOCK
replace PRED with tContract+'7252'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 0
* Pre-Blast (U-R) O/F *
append blank
replace ID with tContract+'7252'+BUILD->BLOCK
replace PRED with tContract+'7251'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 0
* Pre-Blast (INV) O/F *
append blank
replace ID with tContract+'7251'+BUILD->BLOCK
replace PRED with tContract+'7120'+BUILD->BLOCK
replace TYPE with 'FS'
replace LAG with 5
* Assembly for Fabrication *
append blank
replace ID with tContract+'7120'+BUILD->BLOCK
replace PRED with tContract+'7110'+BUILD->BLOCK
replace TYPE with 'SS'
replace LAG with 0

======return

***********************************************************************
* RES1 ------ Creates Resource Data File for Blocks ***************
* * *
* *******
PROC RES1
* Adding RESOURCE Records *
set color to w/#/r
9 17, 30 SAY '---->
set color to w/r
9 16, 30 SAY '---->
selct RES
* Erection *
if BUILD->GRAND = '  
append blank
replace ID with tContract+'7160'+BUILD->BLOCK
replace RESCODE with 'H116'
replace LEVEL with BUILD->BUILD Erect
replace LAVETYPE with 'T1'
append blank
replace ID with tContract+'7160'+BUILD->BLOCK
replace RESCODE with 'P_E'
* replace LEVEL with 1
* replace LAVETYPE with '  
* replace PERIOD with 1
else
append blank
replace ID with tContract+'7150'+BUILD->BLOCK
replace RESCODE with 'H116'
replace LEVEL with BUILD->BUILD Erect
replace LAVETYPE with 'T1'
append blank
replace ID with tContract+'7150'+BUILD->BLOCK
replace RESCODE with 'P_E'
replace LEVEL with 1
replace LEVTYP with 'I'
replace PERIOD with 1

endif

* Post Blast O/F *
append blank
replace ID with tContract+"7254"+BUILD->BLOCK
replace RESCODE with 'H123'
replace LEVEL with BUILD->BUD_OF4
replace LEVTYP with 'I'
append blank
replace ID with tContract+"7254"+BUILD->BLOCK
replace RESCODE with 'P123'
replace LEVEL with 1
replace LEVTYP with 'I'

* Blast & Paint *
append blank
replace ID with tContract+"7253"+BUILD->BLOCK
replace RESCODE with 'H124'
replace LEVEL with BUILD->RND_OF3
replace LEVTYP with 'I'
append blank
replace ID with tContract+"7253"+BUILD->BLOCK
replace RESCODE with 'P124'
replace LEVEL with 1
replace LEVTYP with 'I'

* Pre-Blast (U-R) O/F *
append blank
replace ID with tContract+"7252"+BUILD->BLOCK
replace RESCODE with 'H123'
replace LEVEL with BUILD->BUD_OF2
replace LEVTYP with 'I'
append blank
replace ID with tContract+"7252"+BUILD->BLOCK
replace RESCODE with 'P123'
replace LEVEL with 1
replace LEVTYP with 'I'

* Pre-Blast (INV) O/F *
append blank
replace ID with tContract+"7251"+BUILD->BLOCK
replace RESCODE with 'H123'
replace LEVEL with BUILD->BUD_OF1
replace LEVTYP with 'I'
append blank
replace ID with tContract+"7251"+BUILD->BLOCK
replace RESCODE with 'P123'
replace LEVEL with 1
replace LEVTYP with 'I'

* Assembly *
if BUILD->STRATEGY = 'STD'. or. BUILD->STRATEGY = 'GRAND1'
append blank
replace ID with tContract+"7130"+BUILD->BLOCK
replace RESCODE with 'H113'
replace LEVEL with BUILD->BUD_ASSY
replace LEVTYP with 'I'
append blank
replace ID with tContract+"7130"+BUILD->BLOCK
do case
  case BUILD->ASSY_POS = 'IF'
    replace RESCODE with 'P1131'
  case BUILD->ASSY_POS = 'IC'
    replace RESCODE with 'P1132'
  case BUILD->ASSY_POS = 'IT'
    replace RESCODE with 'P1133'
endcase
replace LEVEL with 1
replace LEVTYP with 'I'
append blank
replace ID with tContract+"7130"+BUILD->BLOCK
replace RESCODE with 'P_A'
replace LEVEL with 1
replace LEVTYP with 'I'
replace PERIOD with 1
replace OFFSET with BUILD->ASSY_D-1

endif
* Sub-Assembly *

replace ID with tContract='7120'+BUILD->BLOCK
replace RESCODE with 'H112'
replace LEVEL with BUILD->BUD_SUB
replace LEVTYPE with 'T'

* Fabrication *
append blank
replace ID with tContract='7110'+BUILD->BLOCK
replace RESCODE with 'H111'
replace LEVEL with BUILD->BUD_FAB
replace LEVTYPE with 'T'

="""return

********************************************************************************
RESG1 --> Creates Resource Data File for Grand Blocks *
********************************************************************************

PROC RESG1
* Adding RESOURCE Records *
set color to w/f/r
@ 17, 30 SAY '----->1
set color to w/f/r
@ 16, 30 SAY '----->1
select RES
* Erection *
append blank
replace ID with tContract='7160'+BUILD->BLOCK
replace RESCODE with 'H114'
replace LEVEL with BUILD->BUD_ERECT
replace LEVTYPE with 'T'
append blank
replace ID with tContract='7160'+BUILD->BLOCK
replace RESCODE with 'P_E1'
replace LEVEL with 1
replace LEVTYPE with '1'
replace PERIOD with 1
* Post Blast Off *
append blank
replace ID with tContract='7254'+BUILD->BLOCK
replace RESCODE with 'H1231'
replace LEVEL with BUILD->BUD_OF4
replace LEVTYPE with 'T'
append blank
replace ID with tContract='7254'+BUILD->BLOCK
replace RESCODE with 'P1231'
replace LEVEL with 1
replace LEVTYPE with '1'

="""return

********************************************************************************
RES2 --> Creates Resource Data File for Blocks *
********************************************************************************

PROC RES2
* Adding RESOURCE Records *
set color to w/f/r
@ 17, 30 SAY '----->1
1349  set color to w+r
1350  @ 16, 30 SAY '---->
1351  select RES
1352  * Erection *
1353  append blank
1354  replace ID with tContract+'7150'+BUILD->BLOCK
1355  replace RESCODE with 'H114'
1356  replace LEVEL with BUILD->BUD_ERECT
1357  replace LEVTYPE with 'T'
1358  append blank
1359  replace ID with tContract+'7150'+BUILD->BLOCK
1360  replace RESCODE with 'P_E'
1361  replace LEVEL with 1
1362  replace LEVTYPE with '1'
1363  replace PERIOD with 1
1364  * Assembly *
1365  append blank
1366  replace ID with tContract+'7130'+BUILD->BLOCK
1367  replace RESCODE with 'H113'
1368  replace LEVEL with BUILD->BUD_ASSY
1369  replace LEVTYPE with 'T'
1370  append blank
1371  replace ID with tContract+'7130'+BUILD->BLOCK
1372  replace RESCODE with 'P1131'
1373  replace LEVEL with 1
1374  replace LEVTYPE with '1'
1375  append blank
1376  replace ID with tContract+'7130'+BUILD->BLOCK
1377  replace RESCODE with 'P_A'
1378  replace LEVEL with 1
1379  replace LEVTYPE with '1'
1380  replace PERIOD with 1
1381  replace OFFSET with BUILD->ASSY_D-1
1382  * sub-Assembly *
1383  append blank
1384  replace ID with tContract+'7120'+BUILD->BLOCK
1385  replace RESCODE with 'H112'
1386  replace LEVEL with BUILD->BUD_SUB
1387  replace LEVTYPE with 'T'
1388  * Fabrication *
1389  append blank
1390  replace ID with tContract+'7110'+BUILD->BLOCK
1391  replace RESCODE with 'H111'
1392  replace LEVEL with BUILD->BUD_FAB
1393  replace LEVTYPE with '1'
1394
1395  return

1396

1397
1398
1399
1400
1401  ************************************************************************
1402  * RESG2 --> Creates Resource Data File for Grand Blocks *
1403  *************
1404  *
1405  ****************************************************************************
1406  PROC RESG2
1407  *
1408  * Adding RESOURCE Records *
1409  set color to w+r
1410  @ 17, 30 SAY '---->
1411  @ 16, 30 SAY '---->
1412  set color to w+r
1413  @ 16, 30 SAY '---->
1414  select RES
1415  * Erection *
1416  append blank
1417  replace ID with tContract+'7160'+BUILD->BLOCK
1418  replace RESCODE with 'H114'
1419  replace LEVEL with BUILD->BUD_ERECT
1420  replace LEVTYPE with 'T'
1421  append blank
1422  replace ID with tContract+'7160'+BUILD->BLOCK
1423  replace RESCODE with 'P_E'
1424  replace LEVEL with 1
1425  replace LEVTYP with '^
1426  replace PERIOD with 1
1427
1428  * Post Blast O/F *
1429  append blank
1430  replace ID with tContract+'7254'+BUILD->BLOCK
1431  replace RESCODE with 'H123'
1432  replace LEVEL with BUILD->BUD_OF4
1433  replace LEVTYP with 'T'
1434  append blank
1435  replace ID with tContract+'7254'+BUILD->BLOCK
1436  replace RESCODE with 'P123'
1437  replace LEVEL with 1
1438  replace LEVTYP with '^
1439  * Blast & Paint *
1440  append blank
1441  replace ID with tContract+'7253'+BUILD->BLOCK
1442  replace RESCODE with 'H124'
1443  replace LEVEL with BUILD->BUD_OF3
1444  replace LEVTYP with 'T'
1445  append blank
1446  replace ID with tContract+'7253'+BUILD->BLOCK
1447  replace RESCODE with 'P124'
1448  replace LEVEL with 1
1449  replace LEVTYP with '^
1450  * Pre-Blast (U-R) O/F *
1451  append blank
1452  replace ID with tContract+'7252'+BUILD->BLOCK
1453  replace RESCODE with 'H125'
1454  replace LEVEL with BUILD->BUD_OF2
1455  replace LEVTYP with 'T'
1456  append blank
1457  replace ID with tContract+'7252'+BUILD->BLOCK
1458  replace RESCODE with 'P125'
1459  replace LEVEL with 1
1460  replace LEVTYP with '^
1461  * Pre-Blast (INV) O/F *
1462  if BUILD->STRATEGY = 'GRAND2' or BUILD->STRATEGY = 'GRAND4' 
1463  append blank
1464  replace ID with tContract+'7251'+BUILD->BLOCK
1465  replace RESCODE with 'H123'
1466  replace LEVEL with BUILD->BUD_OF1
1467  replace LEVTYP with 'T'
1468  append blank
1469  replace ID with tContract+'7251'+BUILD->BLOCK
1470  replace RESCODE with 'P123'
1471  replace LEVEL with 1
1472  replace LEVTYP with '^
1473  endif
1474  * Assembly *
1475  if BUILD->STRATEGY = 'GRAND2' 
1476  append blank
1477  replace ID with tContract+'7130'+BUILD->BLOCK
1478  replace RESCODE with 'H113'
1479  replace LEVEL with BUILD->BUD_ASSY
1480  replace LEVTYP with 'T'
1481  append blank
1482  replace ID with tContract+'7130'+BUILD->BLOCK
1483  do case
1484  case BUILD->ASSY_POS = 'F'
1485    replace RESCODE with 'P1131'
1486    case BUILD->ASSY_POS = 'C'
1487    replace RESCODE with 'P1132'
1488    case BUILD->ASSY_POS = 'T'
1489    replace RESCODE with 'P1133'
1490  end case
1491  replace LEVEL with 1
1492  replace LEVTYP with '^
1493  append blank
1494  replace ID with tContract+'7130'+BUILD->BLOCK
1495  replace RESCODE with 'P_A'
1496  replace LEVEL with 1
1497  replace LEVTYP with '^
1498  replace PERIOD with 1
replace OFFSET with BUILD->ASSY_D-1
endif
<=return

******************************************************************************
* MODEL_M -> Text for Block Build Menu
* *
******************************************************************************
PROC MODEL_M
set color to gr/b
a 4, 17 to 18, 58 double
a 4, 30 SAY " MODEL BUILD MENU "
set color to w/r
a 5, 18 SAY '  

a 6, 18 SAY '  

a 7, 18 SAY ' Enter Contract Letter : :  

a 8, 18 SAY ' Enter Project Name w/Path  

a 9, 18 SAY '  

a 10, 18 SAY '  

a 11, 18 SAY ' Enter Build Data Base w/Path  

a 12, 18 SAY '  

a 13, 18 SAY '  

a 14, 18 SAY '  

a 15, 18 SAY '  

a 16, 18 SAY '  

a 17, 18 SAY '  

set color to gr
a 19, 26 say 'Press "Esc" to Return'  
<=return

*Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:47 AM.
dANALYST found 0 error(s), 0 warning(s), 1540 lines.
INPUT FILE: C:\DPLAN\SP8_DATA.PRG

* FILE NAME: SP8_DATA.PRG
* BY: D. McGauley
* DATE: March 24, 1992
* DESC:
* CALLED BY:
* DATA FILES:
* SP8_DATA.prg

set procedure to SP8_DATA

set space(30) to tAgg
set space(30) to tCurve
set color to gr+/b
a 4, 17 to 18, 58 double
a 4, 30 SAY " CURVE GENERATION ":
set color to w+r
a 5, 18 SAY '
'a 6, 18 SAY '
Enter Aggregation File w/Path
'a 7, 18 SAY '
'a 8, 18 SAY '
'a 9, 18 SAY '
'a 10, 18 SAY '
Enter Curve File w/Path
'a 11, 18 SAY '
'a 12, 18 SAY '
'a 13, 18 SAY '
'a 14, 18 SAY '
'a 15, 18 SAY '
'a 16, 18 SAY '
'a 17, 18 SAY '
set color to gr'
a 19, 26 say 'Press "Esc" to Return'

do while .T.
a 8, 23 get tagg picture '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
read
  if readkey() = 12
    return
  endif
  store trim(tAgg)+.agg to dAgg
  if not. file(dAgg)
    set color to gr+/r
    a 9, 18 SAY 'File NOT Found'
  endif
  loop
  else
  endif
  enddo
set color to gr+/r
a 9, 18 SAY '

---
do while .T.
a 11, 23 get tCurve picture '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
read
  if readkey() = 12
    return
  endif
  store trim(tCurve)+.dbf to dCurve
  if not. file(tCurve)
    set color to gr+/r
    a 12, 18 SAY 'File NOT Found'
  endif
  loop
  else
  endif
  enddo
set color to gr+/r
a 12, 18 SAY '

select B
```plaintext
use &S&Agg index TODATE alias AGG
reindex

select A
use &S&Curve alias CURVE

store cto(' ') to tDate
store 0 to tPeriod
store 0 to tDays
a 13, 18 SAY 'Enter # of Day /Period :

a 14, 18 SAY 'Enter Start Date :

a 15, 18 SAY 'Enter Number of Periods :
do while t...
a 13, 47 get tDays picture '99'
a 14, 47 get tDate
a 15, 47 get tPeriod picture '99'
read
if readkey() = 12
  return
endif
store tDate + 6 to tStart
store (tDate + tPeriod) - 1 to tFinish
select B
seek tStart
if eof()
  a 1A, 18 SAY 'Starting Date
set color to w*/r
  a 1A, 18 SAY 'Not Found'
set color to w*/r
endif
next loop
a 16, 18 SAY '
next exit
enddo

select AGG
copy to SP8_TEMP fields TODATE,WORKDS for recno() < tPeriod+1
set index to RES_DATE
reindex
select CURVE
zap
append from SP8_TEMP
set index to TODATE
reindex
select AGG
set relation to TODATE into CURVE
set color to gr*/r

*** Creating Curve Data ***

a 17, 18 SAY ' Creating Data for Rescode H111
store 'H111' to tRescode
store 'CURVE->H111' to tField
do DATA1

a 17, 18 SAY ' Creating Data for Rescode H112
store 'H112' to tRescode
store 'CURVE->H112' to tField
do DATA1

a 17, 18 SAY ' Creating Data for Rescode H113
store 'H113' to tRescode
store 'CURVE->H113' to tField
do DATA1

a 17, 18 SAY ' Creating Data for Rescode H1132
store 'H1132' to tRescode
```
store 'CURVE->H1132' to tField

150  do DATA1

152  a 17, 18 SAY ' Creating Data for Rescode H1133'

153  store 'H1133' to tRescode

154  store 'CURVE->H1133' to tField

155  do DATA1

156

157  a 17, 18 SAY ' Creating Data for Rescode H1141'

158  store 'H1141' to tRescode

159  store 'CURVE->H1141' to tField

160  do DATA1

161

162  a 17, 18 SAY ' Creating Data for Rescode H1142'

163  store 'H1142' to tRescode

164  store 'CURVE->H1142' to tField

165  do DATA1

166

167  a 17, 18 SAY ' Creating Data for Rescode H12311'

168  store 'H12311' to tRescode

169  store 'CURVE->H12311' to tField

170  do DATA1

171

172  a 17, 18 SAY ' Creating Data for Rescode H12312'

173  store 'H12312' to tRescode

174  store 'CURVE->H12312' to tField

175  do DATA1

176

177  a 17, 18 SAY ' Creating Data for Rescode H1232'

178  store 'H1232' to tRescode

179  store 'CURVE->H1232' to tField

180  do DATA1

181

182  a 17, 18 SAY ' Creating Data for Rescode H124'

183  store 'H124' to tRescode

184  store 'CURVE->H124' to tField

185  do DATA1

186

187  a 17, 18 SAY ' Creating Data for Rescode P1131'

188  store 'P1131' to tRescode

189  store 'CURVE->P1131' to tField

190  do DATA2

191

192  a 17, 18 SAY ' Creating Data for Rescode P1132'

193  store 'P1132' to tRescode

194  store 'CURVE->P1132' to tField

195  do DATA2

196

197  a 17, 18 SAY ' Creating Data for Rescode P1133'

198  store 'P1133' to tRescode

199  store 'CURVE->P1133' to tField

200  do DATA2

201

202  a 17, 18 SAY ' Creating Data for Rescode P12311'

203  store 'P12311' to tRescode

204  store 'CURVE->P12311' to tField

205  do DATA2

206

207  a 17, 18 SAY ' Creating Data for Rescode P12312'

208  store 'P12312' to tRescode

209  store 'CURVE->P12312' to tField

210  do DATA2

211

212  a 17, 18 SAY ' Creating Data for Rescode P1232'

213  store 'P1232' to tRescode

214  store 'CURVE->P1232' to tField

215  do DATA2

216

217  a 17, 18 SAY ' Creating Data for Rescode P124'

218  store 'P124' to tRescode

219  store 'CURVE->P124' to tField

220  do DATA2

221

222  a 17, 10 SAY ' Creating Data for Rescode P.A'

223  store 'P.A' to tRescode
store 'CURVE->P_A1' to tField

* 17, 10 SAY ' Creating Data for Rescode P_E '  
store 'P_E1' to tRescode 
store 'CURVE->P_E' to tField 
do DATA2

               ******************************************************
select CURVE

* 17, 18 SAY ' Creating Data for Rescode H113 '  
replace all H113 with H1131+H1132+H1133

* 17, 18 SAY ' Creating Data for Rescode H114 '  
replace all H114 with H1141+H1142

* 17, 18 SAY ' Creating Data for Rescode H1231 '  
replace all H1231 with H12311+H12312

* 17, 18 SAY ' Creating Data for Rescode H123 '  
replace all H123 with H1231 +H1232

* 17, 18 SAY ' Creating Data for Rescode H11 '  
replace all H11 with H111+H112

* 17, 18 SAY ' Creating Data for Rescode H12 '  
replace all H12 with H123+H124

* 17, 18 SAY ' Creating Data for Rescode H1 '  
replace all H1 with H11+H12

* 17, 18 SAY ' Creating Data for Rescode P113 '  
replace all P113 with P1131+P1132+P1133

* 17, 18 SAY ' Creating Data for Rescode P1231 '  
replace all P1231 with P12311+P12312

* 17, 18 SAY ' Creating Data for Rescode P123 '  
replace all P123 with P1231+P1232

* 17, 18 SAY ' Creating Data for Rescode P12 '  
replace all P12 with P123+P124

******************************************************************************

******************************************************************************

PROC DATA1

store 0 to temp_Man
seek tRescode
if .not. eof()
do while tRescode .and. .not. eof()
   if WORKPS = 0
    else
       replace &tField with temp_Man
   endif
   replace &tField with (LREQTOT / 8) / WORKPS
   store (LREQTOT / 8) / WORKPS to temp_Man
   skip
endo
doto
endo
do while tRescode .and. .not. eof()
   if WORKPS = 0
    ***return

******************************************************************************

******************************************************************************

PROC DATA2

* Calculates Laydown Requirements *

* Calculates Manning Requirements *
PROC DATA2
store 0 to temp_Man
seek tRescode
if .not. eof()
do while RESCODE = tRescode .and. .not. eof()
  if WORKPOS = 0
    replace &tField with temp_Man
  else
    replace &tField with (LREQTOT) / WORKPOS
    store (LREQTOT) / WORKPOS to temp_Man
  endif
  skip
enddo
endif
return
set procedure to SP8_GNT

** Set up print variables **
store chr(027)+"E" to pReset
store chr(027)+"&100" to pPort
store chr(027)+"&160" to pDpi
store chr(027)+"&120" to pBpp
store chr(027)+"&12A" to pLetter
store chr(027)+"&80"+chr(027)+"(s0p16.67h8.4v0s0s3T" to pFont
store chr(027)+"(10U+chr(027)+"(s0p10.00h13.9v0s0s3b11T" to pFont1
store chr(027)+"(11U+chr(027)+"(s0p6.53h18.0v0s0s3b11T" to pFont2
store chr(027)+"(8U+chr(027)+"(s0p12.00h12.0v0s0s0b6T" to pFont3
store chr(027)+"(10U+chr(027)+"(s0p10.00h12.0v0s0s3b1T" to pFont4
store chr(027)+"&83D" to U_on
store chr(027)+"&83F" to U_off

---

do MENU_1
store space(30) to tAct
do while .T.
  set color to w/r
  @ 6, 18 SAY 'Enter Model Name w/Path :'
  @ 7, 18 SAY ' :
  @ 7, 23 get tAct picture '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
read
  if readkey() = 12
    ===============return
  endif
  store trim(tAct)+".act" to tActivity
  store trim(tAct)+".rel" to tResource
  if file(tActivity)
    set color to w/r
    @ 6, 18 SAY ':
    @ 7, 18 SAY ':
    @ 8, 18 SAY ':
    @ 9, 18 SAY ':
    ===============exit
  else
    set color to w/r
    @ 6, 18 SAY 'Can NOT Find File :'
  ===============(oop
  endif
enddo

** Set up data bases **
set color to w/r
@ 16, 18 SAY ' : Setting Up Data Bases:
store "C:\DATA\BASE\BASE\WASSCO.D.dbf" to tCalendar
store "C:\DATA\BASE\BASE\DATES.ndx" to tDX_D1
store "C:\DATA\BASE\BASE\WASSCO.ndx" to tDX_D2
store "C:\DATA\BASE\SP8T_RECORD.dbf" to tT_Record
store "C:\DATA\BASE\SP8T_HOURS.dbf" to tT_Hours
select C
use &tResource alias Resource
set index to ID
set color to w/r
@ 17, 31 SAY 'Re-Indexing:
reindex
set color to w/r
@ 17, 31 SAY ':
select B
* Setting up Memory Variables *
store ctod(' ') to t711s,t711f
store ctod(' ') to t712s,t712f
store ctod(' ') to t713s,t713f
store ctod(' ') to t715s,t715f
store ctod(' ') to t716s,t716f
store 0 to t711d,t712d,t713d,t715d,t716d
store ' ' to tstack

* Outfitting *
store ctod(' ') to t7251s,t7251f
store ctod(' ') to t7252s,t7252f
store ctod(' ') to t7253s,t7253f
store ctod(' ') to t7254s,t7254f
store 0 to t7251d,t7252d,t7253d,t7254d

* Calculating Floats
select Calendar
set index to &ENQX_01
seek t713f
store NASSCOO to Temp_01
seek t7251s
store NASSCOO to Temp_02

* Assembly Float *
store (Temp_02 - Temp_01) - 1 to Assy_Float
seek t7254f
store NASSCOO to Temp_03
if tstack = ' ' seek t716s
```plaintext
store NASSCOD to Temp_04
else
  seek t715
  store NASSCOD to Temp_04
endif

* Assembly Float *
store (Temp_04 - Temp_03) - 1 to OF_Float

*****************************************************
do SCREEN1 & display schedule on screen
*****************************************************

store 'N' to tPrint
set color to gr/y/b
a 22,19 say 'Print Hard Copy of GANTT Chart (Y/N) :

a 22,57 get tPrint picture '1'
read

if tPrint = 'Y'
  set color to w**
  a 23, 33 say 'PRINTING'
  if tBlock > '500'
    select ACTIVITY
      copy to TEMP_Act for ID = tContract .and. C4 = tBlock
      select D
      use TEMP_ACT alias TEMP_Act
      set index to C_BLK_S
      reindex
go top
    set device to print
    a 1, 1 say pReset
    a 1, 1 say pPort
    a 1, 1 say pPipi
    a 1, 1 say pLetter
    do G_PAGE_D
    do G_PAGE_G
    set device to screen
  use
  else
    select ACTIVITY
      copy to TEMP_Act for ID = tContract .and. substr(ID,6,3) = tBlock
      select D
      use TEMP.ACT alias TEMP.Act
      set index to C_BLK.S
      reindex
      go top
    set device to print
    a 1, 1 say pReset
    a 1, 1 say pPort
    a 1, 1 say pPipi
    a 1, 1 say pLetter
    do G_PAGE_A
    set device to screen
  endif
  set color to gr
  a 23, 31 say '1',
endif

***********************
* PROCEDURE G_PAGE_A *
***********************

PROC G_PAGE_A

store 8 to x
store 1 to y
a x,y+10 say pFont2+'Block 'tBlock'+ Build Strategy'+pFont4
a x,y+10 say '1'
x = x + 2
a x, 2 say U_short 'STEEL'
 a x, 17 say 'Start'
```
x = x + 4

store ' to tHEAD1
store ' to tHEAD2
store ' to tHEAD3
store ' to tSHADE1
store ctof(' ') to S_date
store LSDATE-20 to WK_Start
do DATE_BAR
store 45 to y
@ x,y say pFont
@ x,y say p8ipi
@ x,y+32 say chr(201)
@ x,y-31 say replicate(chr(205),100)
@ x,y+69 say chr(187)
x = x + 1
@ x,y+32 say chr(186)
@ x,y say tHEAD1
@ x,y+69 say chr(186)
x = x + 1
@ x,y+32 say chr(186)
@ x,y say tHEAD2
@ x,y+69 say chr(186)
x = x + 1
@ x,y+32 say chr(186)
@ x,y say tHEAD3
@ x,y+69 say chr(186)
x = x + 1

--do while .not. eof()

if D > 0
    @ x,y-32 say chr(186)
    @ x,y say tSHADE1
    @ x,y+69 say chr(186)
x = x + 1
store ' to tErect
@ x,y say tSHADE1

--case subtr(1D,2,4) = '7110'
    store 'Steel Fabrication' to tDescrip
    store chr(220) to tGantt
--case subtr(1D,2,4) = '7120'
    store 'Steel Sub-Assembly' to tDescrip
    store chr(220) to tGantt
--case subtr(1D,2,4) = '7130'
    store 'Steel Assembly' to tDescrip
    store chr(220) to tGantt
--case subtr(1D,2,4) = '7251'
    store '0/F Pre-Blast (Inverted)' to tDescrip
    store chr(207) to tGantt
--case subtr(1D,2,4) = '7259'
    store '0/F Pre-Blast (Up-Right)' to tDescrip
    store chr(209) to tGantt
--case subtr(1D,2,4) = '7253'
    store '0/F Blast & Paint' to tDescrip
    store chr(127) to tGantt
--case subtr(1D,2,4) = '7254'
    store '0/F Post Blast' to tDescrip
    store chr(223) to tGantt
--case subtr(1D,2,4) = '7150'
    store 'Steel Stacking' to tDescrip
    store chr(177) to tGantt
    store chr(004) to tE_Date
    store 'Y' to tErect
--case subtr(1D,2,4) = '7150'
    store 'Steel Erection' to tDescrip
    store chr(177) to tGantt
    store chr(004) to tE_Date
    store 'Y' to tErect

store (LDATE-LSDATE)+1 to tempDur
store (LSDATE-S_date)+1 to tempDur

tGanttD = if(f(mod(tempDur,7) > 0, int(tempDur/7) + 1, int(tempDur/7)))
```plaintext
tStartD = if(mod(tempDur1,7) > 0, int(tempDur1/7) + 1, int(tempDur1/7))
    if tRect = ' '  
        a x,y-32 say chr(186)
        a x,y-30 say tDescrt
        a x,y+tStartD say replicate(tGantt,tGanttD)
        a x,y+tStartD-10 say LSDATE
        a x,y+tStartD+(tGanttD+2) say LFDATE
        a x,y+69 say chr(186)
        x = x + 1
    else
        a x,y-32 say chr(186)
        a x,y-30 say tDescrt
        a x,y+tStartD say '*'
        a x,y+tStartD say replicate(tGantt,tGanttD)
        a x,y+tStartD-10 say LSDATE
        a x,y+tStartD+(tGanttD+2) say LFDATE
        a x,y+69 say chr(186)
        a x,y+((LSDATE-S_date)/7)+1 say '*'
        x = x + 1
    endif
    endif
    skip
```
store 0 to t7251d, t7252d, t7253d, t7254d

- do while substr(1,3) = temp8l and .not. eof()
  - do case
    - case substr(1,2,3) = '711'
      - store 0 to t711d
      - store LDATE to t711s
      - store LFDATE to t711f
    - case substr(1,2,3) = '712'
      - store 0 to t712d
      - store LDATE to t712s
      - store LFDATE to t712f
    - case substr(1,2,3) = '713'
      - store 0 to t713d
      - store LDATE to t713s
      - store LFDATE to t713f
    - case substr(1,2,3) = '715'
      - store 0 to t715d
      - store LDATE to t715s
      - store LFDATE to t715f
      - store 'T' to t8ack
    - case substr(1,2,3) = '716'
      - store 0 to t716d
      - store LDATE to t716s
      - store LFDATE to t716f
    - case substr(1,2,4) = '7251'
      - store 0 to t7251d
      - store LDATE to t7251s
      - store LFDATE to t7251f
    - case substr(1,2,4) = '7252'
      - store 0 to t7252d
      - store LDATE to t7252s
      - store LFDATE to t7252f
    - case substr(1,2,4) = '7253'
      - store 0 to t7253d
      - store LDATE to t7253s
      - store LFDATE to t7253f
    - case substr(1,2,4) = '7254'
      - store 0 to t7254d
      - store LDATE to t7254s
      - store LFDATE to t7254f
  - endcase
  - skip
- enddo

- a x, y=F 10 say pfont1 u_onu pajt Block Number : '+temp8l k=U_off+pFont4
- a x, y=F 10 say ' ' 
- x = x + 2
- a x, 2 say U_onu 'STEEL'
- a x, 17 say 'Start'
- a x, 26 say 'Comp'
- a x, 35 say 'Dur 4 U_off
- a x, 35 say ' ' 
- a x, 40 say U_onu 'OUTFITTING'
- a x, 56 say 'Start'
- a x, 65 say 'Comp'
- a x, 74 say 'Dur 4 U_off
- a x, 74 say ' ' 
- x = x + 1

- a x, 8 say 'Fab(711)'
- if t711d > 0
  - a x, 17 say t711s
  - a x, 26 say t711f
  - a x, 35 say t711d picture '999'
- endif
- a x, 41 say '0/F Inv.(7251)'
- if t7251d > 0
  - a x, 56 say t7251s
  - a x, 65 say t7251f
  - a x, 74 say t7251d picture '999'
- endif
- x = x + 1
- a x, 3 say 'Sub-Assy(712)'
- if t712d > 0
599   a x, 17 say t712s
600   a x, 26 say t712f
601   a x, 35 say t712d picture '999'
602   endif
603   a x, 41 say '0/F Up-R(7252)'
604   if t7252 > 0
605   a x, 56 say t7252s
606   a x, 65 say t7252f
607   a x, 74 say t7252d picture '999'
608   endif
609   x = x + 1
610   a x, 3 say 'Assembly(713)'
611   if t713d > 0
612    a x, 17 say t713s
613    a x, 26 say t713f
614    a x, 35 say t713d picture '999'
615   endif
616   a x, 42 say 'S/B & P(7253)'
617   if t7253d > 0
618    a x, 56 say t7253s
619    a x, 65 say t7253f
620    a x, 74 say t7253d picture '999'
621   endif
622   x = x + 1
623   a x, 3 say 'Stacking(715)'
624   if t715d > 0
625    a x, 17 say t715s
626    a x, 26 say t715f
627    a x, 35 say t715d picture '999'
628   endif
629   a x, 41 say '0/F Post(7254)'
630   if t7254d > 0
631    a x, 56 say t7254s
632    a x, 65 say t7254f
633    a x, 74 say t7254d picture '999'
634   endif
635   x = x + 2
636   ***************************************
637
638 <===RETURN
639
640
641
642  ***************************************
643  *  PROCEDURE G_PAGE_G  *
644   *  for Grand Blocks  *
645  ***************************************
646
647  PROC G_PAGE_G
648
649    ***************************************
650    go top
651    store '1' to THEAD1
652    store '1' to THEAD2
653    store '1' to THEAD3
654    store '1' to TSHADE1
655    store stod('1') to s date
656    store LSDATE-20 to WK_Start
657    do DATE_BAR
658    store 1 to x
659    store 45 to y
660   a x,y-35 say pFont2+'Grand Block '+'G4+' Build Strategy'+pFont
661   a x,y-35 say '  *
662   a x,y say pBmp
663   x = x + 2
664   a x,y-32 say chr(201)
665   a x,y-31 say replicate(chr(205),100)
666   a x,y-69 say chr(187)
667   x = x + 1
668   a x,y-32 say chr(186)
669   a x,y say THEAD1
670   a x,y-69 say chr(186)
671   x = x + 1
672   a x,y-32 say chr(186)
673   a x,y say THEAD2

---print schedule dates---

---prints date bar headings for
  Gantt portion of the Gantt Sched---
do while not eof
  store substr(ID,6,3) to tempblk
  a x,y-32 say chr(186)
  a x,y say tSHADE1
  a x,y+69 say chr(186)
  x = x + 1

  do while substr(ID,6,3) = tempblk
    if D > 0
      store ":" to tErect
      a x,y say tSHADE1
    do case
      case substr(ID,2,4) = "7110" store "Stm" Fabrication to tDecrpt
      store chr(220) to tGantt
      case substr(ID,2,4) = "7120" store "Stm" Sub-Assembly to tDecrpt
      store chr(220) to tGantt
      case substr(ID,2,4) = "7130" store "Stm" Assembly to tDecrpt
      store chr(220) to tGantt
      case substr(ID,2,4) = "7251" store "OF" Pres-Blast (inverted) to tDecrpt
      store chr(207) to tGantt
      case substr(ID,2,4) = "7252" store "OF" Pres-Blast (Up-Right) to tDecrpt
      store chr(209) to tGantt
      case substr(ID,2,4) = "7253" store "OF" Blast & Paint to tDecrpt
      store chr(127) to tGantt
      case substr(ID,2,4) = "7254" store "OF" Post Blast to tDecrpt
      store chr(223) to tGantt
      case substr(ID,2,4) = "7150" store "Stm" Stacking to tDecrpt
      store chr(004) to tE_Date
      store ":" to tErect
      case substr(ID,2,4) = "7160" store "Stm" Fraction to tDecrpt
      store chr(177) to tGantt
      store chr(004) to tE_Date
      store ":" to tErect
    endcase
    if tErect = "":
      a x,y-32 say chr(186)
      a x,y-30 say tDecrpt
      a x,y-((LSDATE-S_date)/7) say replicate (tgantt,int((LFDATE-LSDATE)/7)+1)
      a x,y-((LSDATE-S_date)/7)-9 say LSDATE
      a x,y+((LSDATE-S_date)/7)+int((LFDATE-LSDATE)/7)+2 say LFDATE
      a x,y+69 say chr(186)
      x = x + 1
    else
      a x,y-32 say chr(186)
      a x,y-30 say tDecrpt
      a x,y-((LSDATE-S_date)/7) say "="
      a x,y-((LSDATE-S_date)/7) say replicate (tgantt,int((LFDATE-LSDATE)/7)+1)
      a x,y-((LSDATE-S_date)/7)-9 say LSDATE
      a x,y+((LSDATE-S_date)/7)+int((LFDATE-LSDATE)/7)+2 say LFDATE
      a x,y+69 say chr(186)
      x = x + 1
    endif
  enddo
899  ~ if
900    set color to */r
901    a 5, 41 say '0/F Inv. (7251)'
902  """"""""""""""""""""""""
903    if t7251d > 0
904      a 5, 56 say t7251s
905      a 5, 65 say t7251f
906      a 5, 74 say t7251d picture '999'
907  L  if
908
909    set color to m+/r
910    a 6, 3 say 'sub-Assy(712)'
911    set color to w/r
912    if t712d > 0
913      a 6, 17 say t712s
914      a 6, 26 say t712f
915      a 6, 35 say t712d picture '999'
916  L  if
917
918    set color to uw
919    a 6, 41 say '0/F UP-R(7252)'
920     set color to w+/r
921       if t7252d > 0
922         a 6, 56 say t7252s
923         a 6, 65 say t7252f
924         a 6, 74 say t7252d picture 19991
925  L  if
926
927    set color to n+/r
928    a 7, 3 say 'Asss41Y(713)'
929     set color to 'W/r
930     --if t713d > 0
931       a 7, 17 say t713s
932       a 7, 26 say t713f
933       a 7, 35 say t713d picture 19991
934   - endif
935
936    set color to w+/r
937    a 7, 42 say 'S/B & P(7253)'
938     set color to w/r
939     if t7253d > 0
940       a 7, 56 say t7253s
941       a 7, 65 say t7253f
942       a 7, 74 say t7253d picture '9990
943  L  if
944
945    --if tstack = ' ' set color to w/r
946      a 8, 3 say 'Erection(716)'
947     set color to W/r
948     if t716d > 0
949       a 8, 17 say t716s
950       a 8, 26 say t716f
951       a 8, 35 say t716d picture '999'
952     L  if
953     - else
954      set color to w+/r
955      a 8, 3 say 'Stacking(715)'
956     set color to hi/r
957     if t715d > 0
958       a 8, 17 say t715s
959       a 8, 26 say t715f
960       a 8, 35 say t715d picture '999'
961     L  if
962     - endif
963
964    set color to W+/r
965    a 8, 41 say '0/F Post(7254)'
966    set color to W/r
967    if t7254d > 0
968      a 8, 56 say t7254s
969      a 8, 74 say t7254d picture '999'
970  L  if
971
972    set color to W+/bg
973    a 11, 1 clear to 21,78
974  a 11, 11 say 'HEAD1
975  a 12, 11 say 'HEAD2
976  a 13, 11 say 'HEAD3
977  set color to /bg
978  a 14, 3 say '711'
979  a 15, 3 say '712'
980  a 16, 3 say '713'
981  a 17, 3 say '72510
982  a 18, 3 say '7252'
983  a 19, 3 say '7253'
984  a 20, 3 say '7254'
985  if tStack = '
986    a 21, 3 say '716'
987    else
988    a 21, 3 say '715'
989  L if
990  if t711d > 0
991    store int((t711s-S_date)/7)+12 to tY1
992    store int((t711f-t711s)/7)+2 to dy1
993    store int((t711s-S_date)/7)+1 to tY2
994    set color to w+/g
995    a 14, tY1 say replicate ('-', int((t711f-t711s)/7)+1)
996    set color to /bg
997    a 14, tY1+dy1 say t711f
998  a 14, tY2 say t711s
999 -if
1000  end if
1001  if t712d > 0
1002    store int((t712s-S_date)/7)+12 to tY1
1003    store int((t712f-t712s)/7)+2 to dy1
1004    store int((t712s-S_date)/7)+1 to tY2
1005    set color to u+/g
1006    a 15, tY1 say replicate ('-', int((t712f-t712s)/7)+1)
1007    set color to /bg
1008    a 15, tY1+dy1 say t712f
1009 1 0 \ 1 \ - i f
1011  if t713d > 0
1012    store int((t713s-S_date)/7)+12 to tY1
1013    store int((t713f-t713s)/7)+2 to dy1
1014    store int((t713s-S_date)/7)+1 to tY2
1015    set color to W+/g
1016    a 16, tY1 say replicate ('-', int((t713f-t713s)/7)+1)
1017    set color to /bg
1018    a 16, tY1+dy1 say t713f
1019 1 0 \ 1 \ - i f
1021  if t7251d > 0
1022    store int((t7251s-S_date)/7)+12 to tY1
1023    store int((t7251f-t7251s)/7)+2 to dy1
1024    store int((t7251s-S_date)/7)+1 to tY2
1025    set color to w+/g
1026    a 17, tY1 say replicate ('-', int((t7251f-t7251s)/7)+1)
1027    set color to /bg
1028    a 17, tY1+dy1 say t7251f
1029 1 0 \ 1 \ - i f
1031  if t7252d > 0
1032    store int((t7252s-S_date)/7)+12 to tY1
1033    store int((t7252f-t7252s)/7)+2 to CM
1034    store int((t7252s-S_date)/7)+1 to tY2
1035    set color to w+/9
1036    a 18, tY1 say replicate ('-', int((t7252f-t7252s)/7)+1)
1037    set color to /bg
1038    a 18, tY2 say t7252s
1039 1 0 \ 1 \ - i f
1041  if t7253d > 0
1042    store int((t7253s-S_date)/7)+12 to tY1
1043    store int((t7253f-t7253s)/7)+2 to dy1
1044 1 0 \ 1 \ - i f
1046  if t7254d > 0
1047    store int((t7254s-S_date)/7)+12 to tY1
1048 1 0 \ 1 \ - i f
1049
store int((t7253s-S_date)/7)+1 to tY2
set color to w/g
a 19, tY1 say replicate (**, int((t7253f-t7253s)/7)+1)
set color to /bg
a 19, tY2 say t7253s
6a 19, tY1+dY1 say t7253f
-endif

-if t7254d < O
store int((t7254s-S_date)/7)+12 to tY1
0000store int((t7254f-t7254s)/7)+2 to dY1
store int((t7254s-S_date)/7)+1 to tY2
set color to w+/g
a 20, tY1 say replicate (**, int((t7254f-t7254s)/7)+1)
set color to /bg
a 20, tY2 say t7254s
a 20, tY1+dY1 say t7254f
-endif

if tStack = $ j
store 't715d' to tErectd
store at715s to tErects
store lt715f to tErectf
else
store 't715d' to tErectd
store 't715s' to tErects
store 't715f' to tErectf
endif

if &tErectd > 0
store int((&tErects-S_date)/7)+12 to tY1
store int((&tErectf-&tErects)/7)+2 to dY1
store int((&tErects-S_date)/7)+1 to tY2
set color to w+/g
a 21, tY1 say replicate ('-', int((&tErectf-&tErects)/7)+1)
set color to /bg
a 21, tY2 say &tErects
a 21, tY1+dY1 say &tEsectf

PROC MENU-1

set color to gr+/b
a 4, 17 to 18, 58 double
a 4, 30 SAY "GANTT CHART MENU !!
set color to u+/r
a 5, 18 SAY ' "
a 6, 18 SAY ' "
a 7, 18 SAY ' "
a 8, 18 SAY ' "
a 9, 18 SAY ' "
a 10, 18 SAY ' "
a 11, 18 SAY ' "
a 12, 18 SAY ' "
a 13, 18 SAY ' "
a 14, 18 SAY ' "
a 15, 18 SAY ' "
a 16, 18 SAY ' "
a 17, 18 SAY ' "
set color to gr
a 19, 26 say 'Press l!Esc"to Return'

PROCEDURE UENU_2
* Text for Gantt Hem Format *

PROC MENU_2

set color to gr+/b
a 4, 17 to 18, 58 double

set color to gr+/rl
a 5, 18 SAY ' 1
a 6, 18 SAY ' 1
3 7, 18 SAY 8 Enter Contract Letter : :
3 8, 18 SAY @ Enter Block No. : :
3 9, 18 SAY ' 1
3 10, 18 SAY ' 1
3 11, 18 SAY ' 1
3 12, 18 SAY ' 1
3 13, 18 SAY ' 1
3 14, 18 SAY ' 1
3 15, 18 SAY ' 1
3 16, 18 SAY ' 1
3 17, 18 SAY ' 1
set color to w/r
3 19, 26 say 'Press “Esc” to Return'

set color to gr
a 19, 26 say 'Press “Esc” to Return'

<= return

** Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:48 AM**
dANALYST found 0 error(s), 0 warnin9(S), 153 lines.
set procedure to SP8_LAY

store space(30) to tModel
set color
a 4, 17 to gr+/b
a 4, 30 to gr+/w
set color
da 4, 18 to gr+/r
d 5, 18 to gr+/w
d 6, 18 to gr+/b
d 7, 18 to gr+/r
set color
da 8, 18 to gr+/w
d 9, 18 to gr+/b
da 10, 18 to gr+/r
da 11, 18 to gr+/w
da 12, 18 to gr+/b
da 13, 18 to gr+/r
set color
da 14, 18 to gr+/w
da 15, 18 to gr+/b
da 16, 18 to gr+/r
set color
da 17, 18 to gr+/w
da 18, 18 to gr+/b
set color
da 19, 26 to gr+/t
a 18, 58 to re
SAY "LAYOWN SCHEDULE"
SAY ":
SAY "Enter model Ume w/Path"
SAY ":
SAY ":
SAY ":
set color to gr
da 19, 26 to gr+/t
while .T.
    a 9, 23 get tModel picture
    while .T.
        if readkey() = 12
            if file(tModel)+".act" to dModel
                if file(dModel)
                    set color to w*/r
                    a 15, 18 SAY 'Packing & ReINDEX File'
                else
                    set color to bt+/r
                    a 15, 18 SAY ' File Does Not Exist'
                    set color to b/r
                    i) 15, 26 SAY 'Activity'
                    set color to w/r
            endif
        else
            select A
            use Model index TABLE_LS alias MODEL
            set color to w*/r
            a 15, 18 SAY 'Packing & ReINDEX File'
            set color to n+/r
            a 7, 18 s A Y #
            a 8, 18 s A Y #
            a 9, 18 s A Y #
            a 15, 18 SAY '
            peck
            reindex
            - setting @ the Print Codes **
* "normal_1" Standard Setup, used to print lines,background,gantt
store chr(027)+*(c01+c02)+*(c0p16.67h8.5v0s0b63T+chr(027)+"1&100" to normal_1
* "normal_2" used to print Data Information
store chr(027)+*(c01+c02)+*(c0p16.67h8.4v0s0b63T+chr(027)+"1&100" to normal_2
* "resetP" Reset Printer **
74 store chr(027)+E to reset P
75 * "legel" Set for legel paper ***
76 store chr(027)+8&3A to legel
77 * "PRT_pil6" 6 Lines per inch ***
78 store chr(027)+8&60 to PRT_pil6
79 * "PRT_pil8" 8 Lines per inch ***
80 store chr(027)+8&8D to PRT_pil8
81 * "under_y" Starts underlining print ***
82 store chr(027)+8&DC to under_y
83 * "under_n" Stops underlining print ***
84 store chr(027)+8&3A to under_n
85 * "BOLD_1" used to print Titles
86 store chr(027)+*"(OU+chr(027)+*(s0p10.00h13.9v0s3b11T) to BOLD_1
87 * "BOLD_2" used to print Titles
88 store chr(027)+*"(OU+chr(027)+*(s0p9.11h16.1v0s3b11T) to BOLD_2
89 * "BOLD_3" used to print Titles
90 store chr(027)+*"(OU+chr(027)+*(s0p6.5vh18.0v0s3b11T) to BOLD_3

92 ***********************************************************************

93 * Re-Calculate the Gantt Placements for print-out **
94 set color to w/r,w/r
95 store 'Y' to tPLACE
96 a 7, 18 say ' Recalculating Gantt Placements (Y/N) ' 
97 a 7,56 get tPLACE picture '!' 
98 read

99——if readkey() = 12
100 ———return
101 —endif

102——if tPLACE = 'Y'
103 —set color to w/r,w/r
104 —a 14, 18 say ' Gantt Placements ' 
105 —set color to w/r
106 —a 14, 23 say 'Recalculating' 
107 —do PLACEMENT
108 —set color to w/r
109 —a 14, 18 say ' 
110 —endif

111 ***********************************************************************

112 —do while .T.
113 —store ' ' to t92A,t92B,t93A,t93B
114 —store ' ' to tShop,tTable1,tTable2
115 —store ' ' to tArea_A,tArea_B
116 —set color to gr+r, /w
117 —a 5, 18 SAY ' Steel Outfitting ' 
118 —set color to w/r
119 —a 7, 18 SAY ' Shop Area "A" ' 
120 —a 8, 18 SAY ' Table #1 Area "B" ' 
121 —a 9, 18 SAY ' Table #2 ' 
122 —set color to gr+r
123 —a 12, 18 SAY ' 1992 1993 ' 
124 —set color to w/r
125 —a 13, 18 SAY ' Jan->Jun ' 
126 —a 14, 18 SAY ' Jul->Dec ' 
127 —set color to gr+r
128 —a 16, 18 SAY ' Select with an "X" Your Choices ' 
129 —a 7, 22 get tShop picture '!' 
130 —a 8, 22 get tTable1 picture '!' 
131 —a 9, 22 get tTable2 picture '!' 
132 —a 7, 42 get tArea_A picture '!' 
133 —a 8, 42 get tArea_B picture '!' 
134 —a 13, 37 get t92A picture '!' 
135 —a 14, 37 get t92B picture '!' 
136 —a 13, 42 get t93A picture '!' 
137 —a 14, 42 get t93B picture '!' 

138 —read
139 ———if readkey() = 12
140 —————return
141 ———endif

142 ***********************************************************************

143 —set color to w/r
144 —a 16, 18 SAY ' Setting up Memory Variables ' 
145 —set color to w/r
146 ** Memory Variables to Create Calendar Grid **
```plaintext
set up background variables

---

Set up date bar

heading variables

---

set up print condition variables

---

set color to w+rf
16, 18 SAY 'Printing',

-if 'Shop' = 'X'
  store 'PLANT LAYOUT SCHEDULE' TO title1
  store 'SHOP ASSEMBLY' TO title2
  set color to w+rf
  @ 7, 2+ SAY 'Shop'.
  store 'SHOP' to body
do PUT LAY
set color to w+rf
```
224  &l 7, 24 SAY 'shop'
225  endif
226
227  if table = 'X'
228     store 'PLATEN LAYDOWN SCHEDULE' TO title
229     store 'TABLE #1' to title2
230     set color to w+/r
231     a 8, 24 SAY 'Table #1'
232     store 'TABLE' to body
233     do PRT_LAY
234     set color to W+/r
235     a 8, 24 SAY 'Table #1'
236  endif
237
238  if tTable2 = 'X'
239     store 'PLATEN LAYDOWN SCHEDULE' TO title
240     store 'TABLE #2' to title2
241     set color to w+/r
242     a 9, 24 SAY 'Table #2'
243     store 'TABLE2' to body
244     do PRT_LAY
245     set color to w+/r
246     a 9, 24 SAY 'Table #2'
247  endif
248
249  if tArea_A = 'X'
250     store 'ON-BLOCK LAYDOWN SCHEDULE' TO title
251     store 'AREA 'A" to title2
252     set color to w+/r
253     a 7, 44 SAY 'Area "A"
254     store 'Printing Table #2' to tMEMO1
255     store 'AREA_A' to body
256     do PRT_LAY
257     set color to n+r
258     a 7, 44 SAY 'Area "A"
259  endif
260
261  if tArea_B = 'X'
262     store 'ON-BLOCK LAYDOWN SCHEDULE' TO title
263     store 'AREA 'B" to title2
264     set color to w+/r
265     a 8, 44 SAY 'Area "B"
266     store 'Printing Table #2' to tMEHO1
267     store 'AREA-B' to body
268     do PRT_LAY
269     set color to w+/r
270     a 8, 44 SAY 'Area "B"
271  endif
272
273  return
274
275  *******************************
276  *  PROCEDURE PRT_LAY *
277  **************************************************************************
278  PRW PRT_LAY
279  store O to x
280  store O to y
281  set device to print
282  a x,y say legel
283  a x,y say normal-l
284  a x,y say PRT-1P18
285  set device to screen
286  if t92A = 'X'
287     set device to screen
288     set color to 9r+/r
289     a 12, 35 say '1992'
290     set color to w+/r
291     a 13, 25 say 'Jan'>Jm
299 set device to print
300 store dtln-A92A to LineA
301 store dtln-W2A to LineB
302 store dtln-C92A to LineC
303 store dtln-D92A to LineD
304 store dtln-E92A to LineE
305 store start92A-1 to istart
306 store start92A+182 to icomplete
307 store to title3
308 do heading
309 do &body
310 set device to screen
311 set color to gr+/r
312 a 12, 35 say '1992'
313 set color to Wr
314 a 13, 25 say 'Jan->Jun'
315 —endif
316
317 —if t92B = 'X'
318 set device to screen
319 set color to gr+/r
320 a 12, 35 say '1992'
321 set color to u+/r
322 a 14, 25 say 'July->Dec'
323 set device to print
324 store dtln-A92B to lineA
325 store dtln-B92B to lineB
326 store dtln-C92B to lineC
327 store dtln-D92B to lineD
328 store dtln-E92B to lineE
329 store start92B to istart
330 store start92B+182 to icomplete
331 store to title3
332 do heading
333 do &body
334 set device to screen
335 set color to gr+/r
336 a 12, 35 say 81992:
337 set color to bf+/r
338 a 14, 25 say 'Jul->Dec'
339 —endif
340
341 —if t93A = 'X'
342 set device to screen
343 set color to gr+/r
344 a 12, 40 say '1993'
345 set color to u+/r
346 a 13, 25 say 'Jan->Jun'
347 set device to print
348 store dtln-A93A to lineA
349 store dtln-B93A to lineB
350 store dtln-C93A to lineC
351 store dtln-D93A to lineD
352 store dtln-E93A to lineE
353 store start93A-1 to istart
354 store start93A+182 to icomplete
355 store to title3
356 do heading
357 do &body
358 set device to screen
359 set color to gr+/r
360 a 12, 40 say '1993'
361 set color to wr
362 a 13, 25 say 'Jan->Jun'
363 —endif
364
365 if t93B = 'X'
366 set device to screen
367 set color to gr+/r
368 a 12, 40 say '1993'
369 set color to wr
370 a 14, 25 say 'July->Dec'
371 set device to print
372 store dtln-A93B to lineA
373 store dtln-B93B to lineB
374  store dtln_D93B to lineC
375  store dtln_D93B to lineD
376  store dtln_E93B to lineE
377  store start93B-1 to tSTART
378  store start93B+182 to tCOMPLETE
379  store ' ' to title3
380  do HEADING
381  do &BODY
382  set device to screen
383  set color to gr+/r
384  @ 12, 40 say '1993'
385  set color to w+r
386  @ 14, 25 say 'Jul->Dec'
387  endif
388
389  ====>return
390
391
392
393  **************************************************************************************************
394  *  PROCEDURE HEADING                          *
395  *  Prints Titles & Date Bar Headings            *
396  **************************************************************************************************
397
398  PROC HEADING
399
400  x = 1
401  @ x,y+27 say BOLD_2+title1+normal_1
402  @ x,y+27 say ' '
403  @ x,y+90 say BOLD_3+title2+normal_1
404  @ x,y+90 say ' '
405  x = x+1
406  @ x,y say chr(201)
407  @ x,y+21 say replicate(chr(205),20)
408  @ x,y+21 say chr(203)
409  @ x,y+171 say 'Printed on '+dtoc(date())
410  @ x,y+191 say replicate(chr(205),13)
411  @ x,y+204 say chr(187)
412  x = x+1
413  @ x,y say chr(186)
414  @ x,y+10 say 'MONTH/YEAR'
415  @ x,y+21 say chr(186)
416  @ x,y+22 say lineA
417  @ x,y+204 say chr(186)
418  x = x+1
419  @ x,y say chr(186)
420  @ x,y+10 say ' WEEK NO'
421  @ x,y+21 say chr(186)
422  @ x,y+22 say lineD
423  @ x,y+22 say replicate(lineW,26)
424  @ x,y+204 say chr(186)
425  x = x+1
426  @ x,y say chr(186)
427  @ x,y+21 say chr(186)
428  @ x,y+22 say lineB
429  @ x,y+22 say replicate(lineW,26)
430  @ x,y+204 say chr(186)
431  x = x+1
432  @ x,y say chr(186)
433  @ x,y+10 say ' DAY'
434  @ x,y+21 say chr(186)
435  @ x,y+22 say lineC
436  @ x,y+22 say replicate(lineW,26)
437  @ x,y+204 say chr(186)
438  x = x+1
439  *@ x,y say chr(199)
440  *@ x,y+1 say replicate(chr(196),20)
441  *@ x,y+21 say chr(215)
442  *@ x,y+22 say replicate(chr(196),182)
443  *@ x,y+22 say lineE
444  *@ x,y+204 say chr(182)
445  *x = x+1
446
447  ====>return
PROC PLACEMENT

go top
locate for .not. table = '1'
do while .not. eof()
    store TABLE to tTABLE
    store '1' to tPLACE
do case
case TABLE = 'SH'
do while TABLE = tTABLE
do case
case tPLACE = '1'
    replace PLACEMENT with '1'
    store '2' to tPLACE
    case tPLACE = '2'
    replace PLACEMENT with '2'
    store '3' to tPLACE
    case tPLACE = '3'
    replace PLACEMENT with '3'
    store '4' to tPLACE
    case tPLACE = '4'
    replace PLACEMENT with '4'
    store '5' to tPLACE
    case tPLACE = '5'
    replace PLACEMENT with '5'
    store '6' to tPLACE
    case tPLACE = '6'
    replace PLACEMENT with '6'
    store '7' to tPLACE
    case tPLACE = '7'
    replace PLACEMENT with '7'
    store '8' to tPLACE
    case tPLACE = '8'
    replace PLACEMENT with '8'
    store '9' to tPLACE
    case tPLACE = '9'
    replace PLACEMENT with '9'
    store '10' to tPLACE
    case tPLACE = '10'
    replace PLACEMENT with '10'
    store '1' to tPLACE
endcase
skip
otherwise
    do while TABLE = tTABLE
do case
case tPLACE = '1'
    replace PLACEMENT with '1'
    store '2' to tPLACE
    case tPLACE = '2'
    replace PLACEMENT with '2'
    store '1' to tPLACE
endcase
skip
enddo
endcase
enddo
endcase
return

PROC SHOP
524  store 'SH-00' to tFIND
525  do DIVIDE
526  do LINE
527  store '  ' to title4
528  do GANTT_ln
529  store tCondit1 to t_cond
530  x = x+1
531  store normal_1 to title4
532  do LINE
533  do DATA_ln
534  x = x+1
535  store '  ' to title4
536  do GANTT_ln
537  store tCondit2 to t_cond
538  do LINE
539  do GANTT_ln
540  x = x+1
541  store ' ASSEMBLIES ' to title4
542  do LINE
543  do DATA_ln
544  x = x+1
545  store '  ' to title4
546  do GANTT_ln
547  store tCondit3 to t_cond
548  do LINE
549  do DATA_ln
550  x = x+1
551  store tCondit4 to t_cond
552  do LINE
553  do GANTT_ln
554  x = x+1
555  do LINE
556  do GANTT_ln
557  x = x+1
558  do DATA_ln
559  x = x+1
560  store tCondit5 to t_cond
561  do LINE
562  do GANTT_ln
563  x = x+1
564  do LINE
565  do DATA_ln
566  x = x+1
567  store tCondit6 to t_cond
568  do LINE
569  do GANTT_ln
570  x = x+1
571  do LINE
572  do DATA_ln
573  x = x+1
574  store tCondit7 to t_cond
575  do LINE
576  do GANTT_ln
577  x = x+1
578  do LINE
579  do DATA_ln
580  x = x+1
581  store tCondit8 to t_cond
582  do LINE
583  do GANTT_ln
584  x = x+1
585  do LINE
586  do DATA_ln
587  x = x+1
588  store tCondit9 to t_cond
589  do LINE
590  do GANTT_ln
591  x = x+1
592  do LINE
593  do DATA_ln
594  x = x+1
595  store tCondit10 to t_cond
596  do LINE
597  do GANTT_ln
598  x = x+1
599  do LINE
600  do DATA_in
601  x = x+1
602  do BOTTOM
603  eject
604
605  return
606
607  ********************************************
608  * PROCEDURE TABLE1
609  Table 1 Laydown Format
610  ********************************************
611
612  PROC TABLE1
613
614  *** POSITION 01-01 ***
615  store '01-01' to tFIND
616  * Top Line (Dividing Line) *
617  do DIVIDE
618  * Line 1 (Top Line of Scheduled Work) *
619  store '  ' to title6
620  do LINE
621  store tCondit1 to t_cond
622  do GANIT_in
623  x = x+1
624  * Line 2 (Information Line for Gantt Line #1 )
625  store BOLD_1** 01-01*normal_1 to title6
626  do LINE
627  store tCondit1 to t_cond
628  do DATA_in
629  x = x+1
630  * Line 3 (Line #2 of Gantt Schedule ) *
631  store ' 541 x 601 " to title6
632  do LINE
633  store tCondit2 to t_cond
634  do GANIT_in
635  x = x+1
636  * Line 4 (Information for Gantt Line #2) *
637  store '  ' to title6
638  do LINE
639  store tCondit2 to t_cond
640  do DATA_in
641  x = x+1
642  * Line 5 (Dividing Line) *
643  do DIVIDE
644
645  *** POSITION 01-02 ***
646  store '01-02' to tFIND
647  * Line 1 (Top Line of Scheduled Work) *
648  store '  ' to title6
649  do LINE
650  store tCondit1 to t_cond
651  do GANIT_in
652  x = x+1
653  * Line 2 (Information Line for Gantt Line #1 )
654  store BOLD_1** 01-02*normal_1 to title6
655  do LINE
656  store tCondit1 to t_cond
657  do DATA_in
658  x = x+1
659  * Line 3 (Line #2 of Gantt Schedule ) *
660  store ' 541 x 601 " to title6
661  do LINE
662  store tCondit2 to t_cond
663  do GANIT_in
664  x = x+1
665  * Line 4 (Information for Gantt Line #2) *
666  store '  ' to title6
667  do LINE
668  store tCondit2 to t_cond
669  do DATA_in
670  x = x+1
671  * Line 5 (Dividing Line) *
672  do DIVIDE
*** POSITION 01-03 ***

store '01-03 to tFIND

* Line 1 (Top Line of Scheduled Work) *
store 1 to title4

do LINE
store tCond1 to t_cond

x = x+1

* Line 2 (Information Line for Gantt Line #1 )
store BOLD_l+1 01-03'+ normal_l to title4

do LINE
store tCond1 to t_cond

data_in

x = x+1

* Line 3 (Line #2 of Gantt Schedule ) *
store 54' x 60' 11 to title4

& LINE

store tCond1 to t_cond

do GAHTT-in

x = x+1

* Line 4 (Information for Gantt Line #2) *
store 1 to title4

do LINE
store tCond1 to t_cond

do GAHTT-in

x = x+1

* Line 5 (Dividing Line) *

do DIVIDE

*** POSITION 01-04 ***

store '01-04 to tFIND

* Line 1 (Top Line of Scheduled Work) *
store 1 to title4

do LINE
store tCond1 to t_cond

x = x+1

* Line 2 (Information Line for Gantt Line #1 )
store BOLD_l+1 01-04 '+ normal_l to title4

do LINE
store tCond1 to t_cond

do GAHTT-in

x = x+1

* Line 3 (Line #2 of Gantt Schedule ) *
store 54' x 60' 11 to title4

do LINE
store tCond1 to t_cond

x = x+1

* Line 4 (Information for Gantt Line #2) *
store 1 to title4

do LINE
store tCond1 to t_cond

do GAHTT-in

x = x+1

* Line 5 (Dividing Line) *

do DIVIDE

~ POSITION 01-05 *

store '01-05 to tFIND

* Line 1 (Top Line of Scheduled Work) *
store 1 to title4

do LINE
store tCond1 to t_cond

x = x+1

* Line 2 (Information Line for Gantt Line #1 )
store BOLD_l+1 01-05 '+ normal_l to title4

do LINE
store tCond1 to t_cond

do GAHTT-in

x = x+1

* Line 3 (Line #2 of Gantt Schedule ) *
store 54' x 60' 11 to title4
do LINE
store tCondit2 to t_cond
X = X+1
* Line 4 (Information for Gantt Line #2) *
store a to title4
do LINE
store tCondit2 to t_cond
do DATA_in
X = X+1
* Line 5 (Dividing Line) *
do DIVIDE

\`PDSITION 01-06 ***
store 101-06S to tFINO
* Line 2 (Information Line for Gantt Line #1 )
store BOLO-l+ 01-06 ' normal I to title4
do LINE
store tConditl to t-cond
do GANTT-in
X = X+1
* Line 3 (Line #2 of Gantt Schedule ) *
store II 54' X 60)l to title4
do LINE
store tCondit2 to t_cod
* Line 4 (Information for Gantt Line #2) *
store to title4
do LINE
store tConditl to t-cond
do DATA-in
X = x+1
* Line 5 (Dividing Line) *
do DIVIDE

*** POSITION 01-07 ***
store '01-07'to tFIND
* Line 1 (Top Line of Scheduled Work) *
store I to title4
do LINE
store tConditl to t-cond
do GANTT_in
X = X+1
* Line 2 (Information Line for Gantt Line #1 )
store BOLD_l+ 01-07 normal I to title4
do LINE
store tConditl to t-cond
do DATA_in
x = x+1
* Line 3 (Line #2 of Gantt Schedule ) *
store " 54' x 60, II to title4
do LINE
store tCondit2 to t_cod
do GANTT_in
x = X+1
* Line 4 (Information for Gantt Line #2) *
store to title4
do LINE
store tCondit2 to t_cond
do DATA-in
x = x+1
* Line 5 (Dividing Line) *
do DIVIDE

*** POSITION 01-08 ***
store 101-08S to tFINO
* Line 1 (Top Line of Scheduled Work) *
store ' to title4
do LINE
store tCondit1 to t_cond

do GANTT_In

x = X+1

* Line 2 (Information Line for Gantt Line #1)

store SOLD-1+1 01-08' normal_I to title4

do LINE

store tCondit1 to t_cond

do DATA_In

x = X+1

* Line 3 (Line #2 of Gantt Schedule)*

store " 54' x 60', " to title4

do LINE

store tCondit2 to t_cond

do GANTT_In

x = X+1

* Line 4 (Information for Gantt Line #2)*

store ' to title4

do LINE

store tCondit2 to t_cond

do DATA_In

x = X+1

* Line 5 (Dividing Line)*

do DIVIDE

*** POSITION 01-09 ***

store '01-09' to tFIND

* Line 1 (Top Line of Scheduled Work)*

store ' to title4

do LINE

store tCondit1 to t_cmd

do GANTT-In

x = x+1

* Line 2 (Information Line for Gantt Line #1)

store SOLD-1+01-09'+normal_I to title4

do LINE

store tCondit1 to t_cond

do DATA_In

x = X+1

* Line 3 (Line #2 of Gantt Schedule)*

store " 54' x 60', " to title4

do LINE

store tCondit2 to t_cond

do GANTT_In

x = X+1

* Line 4 (Information for Gantt Line #2)*

store ' to title4

do LINE

store tCondit2 to t_cond

do DATA_In

x = X+1

* Line 5 (Dividing Line)*

do DIVIDE

* POSITION 01-10 *

store '01-10' to tFIND

* Line 1 (Top Line of Scheduled Work)*

store ' to title4

do LINE

store tCondit1 to t_cond

do GANTT_In

x = X+1

* Line 2 (Information Line for Gantt Line #1)

store SOLD-1+01-10'+normal_1 to title4

do LINE

store tCondit1 to t_cond

do DATA_In

x = X+1

* Line 3 (Line #2 of Gantt Schedule)*

store " 54' x 60', " to title4

do LINE

store tCondit2 to t_cond

do GANTT_In

x = X+1

* Line 4 (Information for Gantt Line #2)*
PROC TABLE2

*** POSITION 02-01 ***
store '02-01' to tFIND
* Top Line (Dividing Line) *
do DIVIDE
* Line 1 (Top Line of Scheduled Work) *
store ' ' to title4
do LINE
store tCondit1 to t_cond
do GANTT_in
x = x+1
* Line 2 (Information Line for Gantt Line #1)
store BOLD_1++ 02-01++normal_1 to title4
do LINE
store tCondit1 to t_cond
do DATA_in
x = x+1
* Line 3 (Line #2 of Gantt Schedule ) *
store " 40' x 50' " to title4
do LINE
store tCondit2 to t_cond
do GANTT_in
x = x+1
* Line 4 (Information for Gantt Line #2) *
store ' ' to title4
do LINE
store tCondit2 to t_cond
do DATA_in
x = x+1
* Line 5 (Dividing Line) *
do DIVIDE

*** POSITION 02-02 ***
store '02-02' to tFIND
* Line 1 (Top Line of Scheduled Work) *
store ' ' to title4
do LINE
store tCondit2 to t_cond
do DATA_in
x = x+1
* Line 2 (Information Line for Gantt Line #1)
store BOLD_1++ 02-02++normal_1 to title4
do LINE
store tCondit1 to t_cond
do DATA_in
x = x+1
* Line 3 (Line #2 of Gantt Schedule ) *
store " 40' x 50' " to title4
do LINE
store tCondit2 to t_cond
do GANTT_in
x = x+1
* Line 4 (Information for Gantt Line #2) *
store ' ' to title4
do LINE
store tCondit2 to t_cond
974  do DATA-in
975  x = X+1
976  * Line 5 (Dividing Line) *
977  do DIVIDE
978
979  * POSITION 02-03 ***
980  store '02-03' to tFIND
981  * Line 1 (Top Line of Schedule Work) *
982  store ':' to title4
983  do LINE
984  store tCond1t to t_cond
985  do GANTT_in
986  x = X+1
987  * Line 2 (Information Line for Gantt Line #1) *
988  store BOLD-l+l 02-03' +normal l to title4
989  do LINE
990  store tCond1t to t_cond
991  do DATA-in
992  x = X+1
993  * Line 3 (Line #2 of Gantt Schedule) *
994  store ' 40' x 50' ' to title4
995  do LINE
996  store tCond1t2 to tCond
997  do GANTT_in
998  x = X+1
999  * Line 4 (Information for Gantt Line #2) *
1000  store ' ' to title4
1001  do LINE
1002  store tCond1t2 to tCond
1003  do DATA-in
1004  x = X+1
1005  * Line 5 (Dividing Line) *
1006  do DIVIDE
1007
1008  *** POSITION 02-04 ***
1009  store '02-04' to tFIND
1010  * Line 1 (Top Line of Schedule Work) *
1011  store : to title4
1012  do LINE
1013  store tCond1t to t_cond
1014  do GANTT_in
1015  x = X+1
1016  * Line 2 (Information Line for Gantt Line #1) *
1017  store BOLD-1+'02-04'+normal l to title4
1018  do LINE
1019  store tCond1t2 to tCond
1020  do DATA-in
1021  x = X+1
1022  * Line 3 (Line #2 of Gantt Schedule) *
1023  store ' 40' x 50' ' to title4
1024  do LINE
1025  store tCond1t2 to tCond
1026  do GANTT_in
1027  x = X+1
1028  * Line 4 (Information for Gantt Line #2) *
1029  store ' ' to title4
1030  do LINE
1031  store tCond1t2 to tCond
1032  do DATA-in
1033  x = X+1
1034  * Line 5 (Dividing Line) *
1035  do DIVIDE
1036
1037
1038  *** POSITION 02-05 ***
1039  store '02-05' to tFIND
1040  * Line 1 (Top Line of Schedule Work) *
1041  store ':' to title4
1042  do LINE
1043  store tCond1t to tCond
1044  do GANTT_in
1045  x = X+1
1046  * Line 2 (Information Line for Gantt Line #1) *
1047  store BOLD-l+'02-05'+normal l to title4
1048  do LINE
store tCondit to t-cond
1050 do DATA-in
1051 x = X+1
1052 " Line 3 (Line #2 of Gantt Schedule ) *
1053 store " 40' x 50' " to title4
1054 do LINE
1055 store tCondit2 to t-cond
1056 do GANTT-in
1057 x = X+1
1058 " Line 4 (Information for Gantt Line #2) *
1059 store to title4
1060 do LINE
1061 store tCondit2 to t-cond
1062 do DATA-in
1063 x = X+1
1064 " Line 5 (Bottom Line) *
1065 do BOTTOM
1066 eject
1067
1068 =return
1069
PROC AREA_A
1070 **********************************************************
1071 *** POSITION A-1 ***
1072 store A-01 to tFIND
1073 do DIVIDE
1074 store to title4
1075 do LINE
1076 store tCondit to t-cond
1077 do GANTT-in
1078 x = X+1
1079 store BOLD-1+ ' A-1 '+normal to title4
1080 do LIME
1081 store tCondit to t-cond
1082 do DATA-in
1083 x = x+1
1084 store 40 x 60' " to title4
1085 do LINE
1086 store tCondit2 to t-cond
1087 do GANTT-in
1088 x = X+1
1089 store ' to title4
1090 do LIME
1091 store tCondit2 to t-cond
1092 do DATA-in
1093 x = x+1
1094 store BOLD-1+ A-2 '+normal 1 to title4
1095 do LINE
1096 store tCondit2 to t-cond
1097 do DATA-in
1098 x = X+1
1099 do DIVIDE
1100 *** POSITION A-2 ***
1101 store ' A-02 ' to tFIND
1102 do LINE
1103 store tCondit to t-cond
1104 do GANTT-in
1105 x = x+1
1106 store BOLD-1+ A-2 '+normal 1 to title4
1107 do LINE
1108 store tCondit2 to t-cond
1109 do DATA-in
1110 x = X+1
1111 store 40' x 60' " to title4
1112 do LINE
1113 store tCondit2 to t-cond
1114 do GANTT_in
1115 x = X+1
1116 do LIME
1117 store tCondit2 to t-cond
1118 do GANTT_in
1119 x = X+1
1120 store ' to title4
1121 do LIME
1122 store tCondit2 to t-cond
1123 do DATA-in
x = X+l

do DIVIDE

*** POSITION A-3 ***

store ' A-03' to tFIND
store ' ' to title4
do LINE
store tConditl to t_cond
do GANTT-in
x = X+l
store BOLD_l+ ' A.3 l+normal l to title4
do LINE
store tcondit1 to t-cond
do DATA-in
x = X+l
store " 40' x 60' " to title4
do LINE
store tCondit2 to t_cond
do GANTT-ln
x = X+l
store BOLD_l+ ' A.4 l+normal to title4
do LINE
store tCondit1 to t_cond
do DATA-in
x = X+l
store " 40' x 60' " to title4
do LINE
store tcondit2 to t-cond
x = X+l
store ' ' to title4
do LINE
store tCondit2 to t_cond
x = X+l
store ' ' to title4
do LINE
store tCondit2 to t_cond
x = X+l
store ' ' to title4
do DIVIDE

*** POSITION A-4 ***

store A-04' to tFIND
store ' ' to title4
do LINE
store tConditl to t_cond
x = X+l
store BOLD_l+ ' A.4 l+normal to title4
do LINE
store tCondit2 to t_cond
do GANTT-in
x = X+l
store " 40' x 60' " to title4
do LINE
store tcondit2 to t-cond
x = X+l
store ' ' to title4
do LINE
store tCondit2 to t_cond
x = X+l
store ' ' to title4
do DIVIDE

*** POSITION A-5 ***

store A-05' to tFIND
store ' ' to title4
do LINE
store tConditl to t_cond
x = X+l
store BOLD_l+ ' A-5 l+normal to title4
do LINE
store tCondit1 to t_cond
x = X+l
store ' 40' x 60' " to title4
do LINE
store tCondit2 to t_cond
x = X+l
store ' ' to title4
do LINE
store tCondit2 to t_cond
x = X+l
store ' ' to title4
do DIVIDE
*** POSITION A-6 ***
store ' A-(06' to tFIND
store ' ' to title4
do LINE
store tcondit1 to t-cond
do GANTT_in
x = X+1
store BOLD-1+ A-6 l+normal to title4
do LINE
store tCondit 1 to t_cond
do DATA_in
x = X+1
store 40' x60' to title4
do LINE
store tCondit2 to t_cond
do GANTT-in
x = X+1
store to title4
do LINE
store tCondit2 to t-cond
do DATA-in
x = X+1
do DIVIDE
*** POSITION A-7 ***
store ' A-07' to tFIND
store # to title4
do LINE
store tcondit1 to t-cond
do GANTT_in
x = X+1
store BOLD-1+ A.7 l+normal 1 to title4
do LINE
store tCondit1 to t-cond
do DATA-in
x = X+1
store 30' X60' to title4
do LINE
store tCondit2 to t_cond
do GANTT_in
x = X+1
store to title4
do LINE
store tCondit2 to t-cond
do DATA-in
x = X+1
do DIVIDE
*** POSITION A-8 ***
store ' A-08' to tFIND
store to title4
do LINE
store tcondit1 to t-cond
do GANTT_in
x = X+1
store BOLD-1+ A-8 l+normal to title4
do LINE
store tCondit1 to t-cond
do DATA-in
x = X+1
store 40' x60' " to title4
do LINE
store tCondit2 to t-cond
do GANTT-in
x = X+1
store to title4
do LINE
store tCondit2 to t-cond
do DATA-in
x = X+1
do BOTTOM
eject
<=.return
*** POSITION B-1 ***
store B-01 'to tFIND
do DIVIDE
to title4
do LINE
store tCond1 to t-cond
do GANTT_In
x = X+1
store BOLD_l+ B-1 'normal_l to title4
do LINE
store tCond1 to t-cond
do DATA_In
x = X+1
store 40' x 60' to title4
do LINE
store tCond2 to t-cond
do GANTT_In
x = X+1
store ' to title4
do LINE
store tCond2 to t-cond
do DATA_In
x = X+1
do DIVIDE
*** POSITION B-2 ***
store E-02' to tFIND
store to title4
do LINE
store tCond1 to t-cond
do GANTT_In
x = X+1
store BOLD-l+ B-2 l+normal to title4
do LINE
store tCond1 to t-cond
do DATA_In
x = X+1
store 40' x 60' to title4
do LINE
store tCond2 to t-cond
do GANTT_In
x = X+1
store to title4
do LINE
store tCond2 to t-cond
do DATA_In
x = X+1
do DIVIDE
*** POSITION B-3 ***
store B-03 to tFIND
store to title4
do LINE
store tCond1 to t-cond
do GANTT_In
x = X+1
store BOLD_l+ B-3 l+normal l to title4
do LINE
store tCond1 to t-cond
do DATA_In
x = X+1
store 40' x 60' to title4
do LINE
store tCond2 to t-cond
do GANTT_In
x = X+1
store to title4
do LIME
store tCondit2 to t-cond
do DATA-in
x = X+1
do DIVIDE

*** POSITION B-4 ***
store B-04' to tFIND
store to title4
do LIME
store tCondit1 to t-cond
do GANTT-in
x = X+1
store BOLD-1+ B.4 l+normal l to title4
do LINE
store tCondit1 to t-cond
do DATA-in
x = X+1
store  40' x 60'  to title4
do LINE
store tCondit2 to t-cond
do GANTT_in
x = X+1
store to title4
store tCondit1 to t-cond
do GANTT-in
x = X+1
store 40' x 60' ** to title4
store tCondit2 to t-cond
do GANTT_in
x = X+1
store BOLD-1+ B-5 l+normal 1 to title4
do LINE
store tCondit1 to t-cond
do DATA-in
x = X+1
store B-05' to tFIND
store tCondit1 to t-cond
do GANTT_in
x = X+1
store 40' x 60' ** to title4
store tCondit2 to t-cond
do GANTT_in
x = X+1
store tCondit1 to t-cond
do DATA-in
x = X+1
store 40' x 60' ** to title4
store tCondit2 to t-cond
do GANTT_in
x = X+1
store 1 to title4
store tCondit2 to t-cond
do DATA-in

*** POSITION B-5 ***
store B-05' to tFIND
store to title4
do LINE
store tCondit1 to t-cond
do GANTT_in
x = X+1
store BOLD-1+ B-5 l+normal 1 to title4
do LINE
store tCondit1 to t-cond
do DATA-in
x = X+1
store 40' x 60' ** to title4
store tCondit2 to t-cond
do GANTT_in
x = X+1
store tCondit1 to t-cond
do DATA-in
x = X+1
store 40' x 60' ** to title4
store tCondit2 to t-cond
do GANTT_in
x = X+1
store 1 to title4
store tCondit2 to t-cond
do DATA-in

*** POSITION B-6 ***
store B-06' to tFIND
store to title4
do LINE
store tCondit1 to t-cond
do GANTT_in
x = X+1
store BOLD-1+ B-6 l+normal 1 to title4
do LINE
store tCondit2 to t-cond
do GANTT_in
x = X+1
store tCondit1 to t-cond
do DATA-in
x = X+1
store 40' x 60' ** to title4
x = X+1

do DIVIDE

*** POSITION B-7 ***
store ' B-07' to tFIND
store ' l to title4

do LINE
store tCond1l to t-cond
do GANTT-in
x = X+1
store BOLD_4+ B-7 l+normal l to title4
do LINE
store tCond1l to t-cond
do DATA-in
x = X+1
store ' 40', x 60' 1 to title4
do LINE
store tCond1l to t-cond
do GANTT-in
x = X+1
store ' l to title4
do LINE
store tCond1l to t-cond
do DATA-in
x = X+1
store ' 40', x 60' 1 to title4
do LINE
store tCond1l to t-cond
store ' l to title4
do LINE
store tCond1l to t-cond
store ' l to title4
do LINE
store tCond1l to t-cond
do DATA-in
x = X+1
store ' ' to title4
do DATA-in
x = X+1
store ' 40', x 60' ' to title4
do LINE
store tCond1l to t-cond
do GANTT-in
x = X+1
store ' ' to title4
do LINE
store tCond1l to t-cond
do DATA-in
x = X+1
do BOTTOM
eject

===> return

===============================================

* PROCEDURE BOTTOM *
* Prints Bottom Line of Format *
===============================================

PROC BOTTOM

a x,y say chr(200)
a x,y+1 say replicate(chr(205),20)
a x,y+21 say chr(202)
a x,y+22 say replicate(chr(205),8)
a x,y+30 say ' X is W/V WELL PLANNED '
a x,y+72 say replicate(chr(205),132)
a x,y+204 say chr(188)
x = x+1
do LEGEND
do BOTTOM

===> return

1498
PROCEDURE GANTT In

Prints a Gantt Bar Line in a Laydoun Format

PROC GANTT In

1508 go top
1509 store ' ' to tGO
1510 SEEK tFIND
1511 if .not. eof()
1512   do while TABLE = tFIND
1513      if LDATE > tSTART .and. LSDATE < tCOMPLETE
1514         store 'Y' to tGO
1515      <===exit
1516    else
1517       if LDATE > tCOMPLETE
1518          <===exit
1519         skip
1520    endif
1521  enddo
1522 if tGO = 'Y'
1523   do while LDATE > tSTART .and. LSDATE < tCOMPLETE .and. TABLE = tFIND
1524      if LDATE <= LSDATE
1525         if &t_cond
1526            do case
1527               case substr(ID,2,2) = '71'
1528                  store chr(219) to tGANTT
1529               case substr(ID,2,4) = '7251'
1530                  store chr(202) to tGANTT
1531               case substr(ID,2,4) = '7252'
1532                  store chr(203) to tGANTT
1533               case substr(ID,2,4) = '7254'
1534                  store chr(178) to tGANTT
1535            endcase
1536         endif
1537      endif
1538      if LDATE < tCOMPLETE
1539         @ x,y+22+(LSDATE-tSTART) say replicate(tGANTT,LSDATE-LSDATE+1)
1540            if tCOMPLETE-LSDATE < 6
1541               @ x,y+22+(183) say SIZE
1542            else
1543               @ x,y+22+(LSDATE-tSTART) say replicate(tGANTT,tCOMPLETE-LSDATE)
1544            endif
1545         endif
1546         @ x,y+22 say replicate(tGANTT,LDATE-tSTART)
1547         @ x,y+22+(LDATE-tSTART) say SIZE
1548      endif
1549      endif
1550  enddo
1551 <===return

PROCEDURE DATA_In

Prints a Data Information Line in a Laydoun Format

PROC DATA_In

goto top
PROC DIVIDE

a x,y  say chr(204)
a x, Y+1 say replicate(chr(205),20)
a x, Y+21 say chr(206)
a x,Y+22 say replicate(chr(205),182)
a x,Y+22 say LineE
a x, Y+204 say chr(185)
X = x+1

PROCEDURE LEGEND

* Prints the legend in a Laydown Format *

PROC LEGEND

store chr(219) to tASSY
store chr(202) to tOF_INV
store chr(203) to tOF_UPR
store chr(176) to tOF_POST
a x,y*8 say 'LEGEND:'
x = x + 1
a x,y*10 say 'Assembly: ' + replicate(tASSY,5)+';'
a x,y*27 say 'O/F Pre-Blast Inverted: ' + replicate(tOF_INV,5)+';'
a x,y*58 say 'O/F Pre-Blast UpRight: ' + replicate(tOF_UPR,5)+';'
a x,y*88 say 'O/F Post Blast: ' + replicate(tOF_POST,5)+';'

<==return

1683
1684
1685
1686
1687
1688

** Formatted by: dANALYST Ver. 7.3a on March 24, 1992 at 9:50 AM.**

dANALYST found 0 error(s), 0 warning(s), 1688 lines.

All files processed.

dANALYST found 0 total error(s), 0 total Warning, 5354 total lines.
APPENDIX VI

APPLICATION OF PC-BASED FACILITIES SIMULATION
Application of PC-Based Facilities Simulation

This report has described the development of a PC-based model which serves as a tool to assist ship yards in developing, updating, and revising schedules, manning, and facility utilization reports. To this point, the report assumed schedules are to be developed based upon a yard's current facility constraints. A shipyard conducting long range planning may wish to look at strategies and schedules based upon a modification to or a different usage of a yard’s facilities. What is the best tool to perform this analysis? A team was formed to investigate the applicability of using a stochastic simulation software package to perform this task. This appendix discusses the insights gained by that group.

The first task the group undertook was to address the questions shown below.

- Is the high level manufacturing process that we are focusing on appropriate for stochastic simulation and, if simulated, are we likely to learn anything that we do not already know?

- If simulating the manufacturing process at this level is not expected to yield useful information, are there other opportunities to take advantage of simulation’s strengths that would add value to the planning process within a shipyard?

These questions are answered relative to the capabilities of the Integrated Production Planning System (IPPS). The strengths of stochastic simulation lies in its ability to:

- Assign finite resources and reschedule accordingly.

- Utilize a stochastic analysis methodology to account for variations in actual performance.

- System is table driven and, therefore, is well suited to conducting a high number of iterations, what-ifs, and sensitivity studies with various facility scenarios.
• Capability to allow observation of the dynamic performance of a simulated manufacturing process.

In contrast, the IPPS provides the capability to develop and utilize a deterministic model. Although finite resources are assigned and reported against, the system is not used to automatically adjust schedules based on that information.

The output of any computer system, whether it be stochastic simulation or deterministic modeling is dependent upon the input given to the system. A stochastic simulation program requires as its input data regarding statistical distribution of activity durations, statistical distributions of performance criteria (i.e., machine downtime), and a set of prioritization rules that take into account all the factors to be considered in schedule development.

There is a high degree of uncertainty associated with long range planning analysis. When analyzing facilities and work flow strategies that do not currently exist, no historical data is available. Performance of individual pieces of equipment at other locations may be investigated but this may not be completely applicable to the system arrangement you are analyzing. With regards to long range planning, broad assumptions regarding statistical distributions of activity durations and performance criteria that are required for stochastic simulation model may be no more meaningful that the more generalized assumptions required by a deterministic model. Greater detail in analysis probably would not have caused a change in the recommendation with respect to facilities modifications.

It is important to understand the relative benefits of finite vs. infinite capacity systems. The finite capacity model manages a high number of complex resource constraints much more efficiently than one could manage with an infinite capacity system. The downside is that the model becomes very complex when trying to establish a realistic prioritization logic. Prioritization rules
and logic are not always known beforehand. Sequencing and constraints are often varied as assumptions are changed in the process of schedule development. The complexity is again increased when manpower constraints are variable. For example, additional shifts or overtime decisions (which decrease activity duration) can be made based upon task priority. Output is generated indicating resource utilization relative to forward and backward pass. Prioritization is left to the discretion of the planner. The planner manually adjusts the model assumptions as the schedule is refined. After a number of iterations are performed to effectively level the manufacturing process, system output will reflect resource utilization projections as well as a working schedule. However, capacity/schedule analysis becomes quite complex and time consuming as more and more resource constraints are added.

Based upon the insights gained by the group, the following recommendation is made in regards to the use of stochastic simulation for long range planning.

The dubious benefits associated with the increased detail of a simulation analysis along with the difficulty of defining a complete set of prioritization rules and logic for finite capacity analysis are not worth the increased training, development, and maintenance effort required. Given the data available and broad assumptions that must be made in a long range planning analysis, equally useful information may be gathered through the use of a more simple, deterministic model.

This does not mean, however, that stochastic simulation has no use in a shipyard environment. If a detailed schedule is being developed for a well defined process (ie. a panel line or a machine shop) stochastic simulation may prove a powerful tool. At this level, with established equipment for which historical data is available, required input of activity durations and down times with their statistical distributions is available. Prioritization rules and logic are clearer at a more detailed level. The rules and logic may be more easily incorporated into the software system so that the software can help produce more meaningful results.
Performing stochastic simulation is an excellent method for resource, capacity, and throughput optimization studies. Simulation can serve as a powerful tool for shipyards. However, like all tools, it is most effective when used properly.
SYSTEM’S DEMONSTRATION DISK

INTEGRATED PRODUCTION PLANNING SYSTEM
SYSTEM'S DEMONSTRATION **DISK**

The "System's Demonstration Disk" is an on screen slide show which gives a graphical overview of the Integrated production Planning System (IPPS). The demonstration disk will step the viewer through both the Open Plan screens used by the IPPS and the screens created by the dBase programs included within this project. To view the demonstration disk, insert the disk into the "B" drive of your computer and type B:\DEMO. Note, the disk requires a VGA monitor to operate.
OF THIS REPORT!

PLEASE RETURN A RESPONSE CARD AFTER READING REPORT.

NSRP READER RESPONSE CARD

We would appreciate your comments on this report. Please take a few minutes to complete and return this postage-paid card. Thank you.

Name ____________________________
Organization ______________________
Phone ____________________________

☑ Overall Quality of Report
☐ Excellent  ☐ Good  ☐ Fair  ☐ Poor

☐ Did/Will your organization implement the results of this project? ☐ Yes ☐ No

If not, why? __________________________________________

How Did You Receive Report?
☐ Mailed directly to you
☐ Referred to you by someone else

Did/Will You Pass Report On To Someone Else?
☐ Yes ☐ No

In Your Opinion, Is Anything Missing That Would Make This Report Better?

☐ Yes______________________________

☐ Genera/ Comments

______________________________

NSRP READER RESPONSE CARD

We would appreciate your comments on this report. Please take a few minutes to complete and return this postage-paid card. Thank you.

Name ____________________________
Organization ______________________
Phone ____________________________

☑ Overall Quality of Report
☐ Excellent  ☐ Good  ☐ Fair  ☐ Poor

☐ Did/Will your organization implement the results of this project? ☐ Yes ☐ No

How Did You Receive Report?
☐ Mailed directly to you
☐ Referred to you by someone else

Did/Will You Pass Report On To Someone Else?
☐ Yes ☐ No

In Your Opinion, Is Anything Missing That Would Make This Report Better?

☐ Yes______________________________

☐ Genera/ Comments

______________________________
Additional copies of this report can be obtained from the National Shipbuilding Research Program Coordinator of the Bibliography of Publications and Microfiche Index. You can call or write to the address or phone number listed below.

NSRP Coordinator
The University of Michigan
Transportation Research Institute
Marine Systems Division
290 Baxter Road
Ann Arbor, MI 48109-2150

Phone: (313) 763-2465
Fax: (313) 936-1081