RASTER GRAPHICS DISPLAY LIBRARY

A Final Report of
NASA Grant NAG-1-391
Presented to

NASA Langley Research Center
ACD M/S 124A
Hampton, Virginia 23665

by

Anders Grimsrud
Graduate Research Assistant

Michael B. Stephenson, Ph.D.
Associate Professor

30 June 1987

Engineering Computer Graphics Laboratory
Civil Engineering Department
Brigham Young University
Provo, Utah 84602
(801) 378-2812

Report No. ECGL-87-03
DISCLAIMER
for
RASTER GRAPHICS DISPLAY LIBRARY

Neither Brigham Young University nor their employees makes any warranty expressed or implied, or assumes any legal responsibility for the accuracy, completeness or usefulness of the computer programs described in this document.
# Table of Contents

## Alphabetical List of Subroutines

### List of Routines by Function

- Groups and Items Routines
- Animation Routines
- Display Device Routines
- Transformation Routines
- Text Routines
- Vector and Scalar Function Routines
- Input/Output Routines
- Raster Routines
- Hidden Line Removal Routines
- Normal Creation and Manipulation Routines
- Clipping Routines
- Input/Output File Routines
- Initialization Routine (Must be included in all user applications)
- Other Routines

## Introduction to the Raster Graphics Display Library

- Raster Display Systems
- A Common Problem of Raster Display Systems
- The Raster Graphics Display Library
- How To Use This Manual

## Chapter Summaries

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example Problems</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>Technical References</td>
<td>1.1</td>
</tr>
<tr>
<td>3</td>
<td>Common Block References</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>Include File References</td>
<td>1.1</td>
</tr>
<tr>
<td>A</td>
<td>Link Map</td>
<td>1.1</td>
</tr>
<tr>
<td>B</td>
<td>An Introduction to Hierarchical Data Structures</td>
<td>1.1</td>
</tr>
<tr>
<td>C</td>
<td>Software Installation</td>
<td>1.1</td>
</tr>
</tbody>
</table>
CHAPTER TWO
TECHNICAL REFERENCES
See pages iii and vii for indices of the routines.

CHAPTER THREE
COMMON BLOCKS
CATTRI ........................................ 3.1
CFLAGS ........................................ 3.2
CHFWAS ......................................... 3.4
CLIMIT .......................................... 3.5
CMENRY ......................................... 3.7
HIDSTF .......................................... 3.9
KEEP ........................................... 3.10
MASTER .......................................... 3.11
PIXSTF .......................................... 3.12
RESOLT .......................................... 3.13
VEWSTF .......................................... 3.14
ZBUFER ........................................ 3.15

CHAPTER FOUR
INCLUDE FILES
ANIM.INC ........................................ 4.1
CVER.INC ........................................ 4.2
FNCT.INC ........................................ 4.3
HIDN.INC ........................................ 4.4
MOVL.INC ........................................ 4.5
MSTR.INC ........................................ 4.6
USER.INC ........................................ 4.7

APPENDIX A
LINK MAP ........................................ A.1

APPENDIX B
AN INTRODUCTION TO HIERARCHICAL DATA STRUCTURES
Hierarchical Data Structure Terms .................. B.1
What Are Groups And Items? ...................... B.2

APPENDIX C
SOFTWARE INSTALLATION ....................... C.1
Installation ................................... C.2
Bugs, Problems, and Comments ................... C.2
<table>
<thead>
<tr>
<th>ALPHABETIZED LIST OF SUBROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTGRP</td>
</tr>
<tr>
<td>ADDCAL</td>
</tr>
<tr>
<td>ANFILI</td>
</tr>
<tr>
<td>ANFILS</td>
</tr>
<tr>
<td>ANIDRV</td>
</tr>
<tr>
<td>AROCLR</td>
</tr>
<tr>
<td>ATFROM</td>
</tr>
<tr>
<td>AUTOON</td>
</tr>
<tr>
<td>BEGANI</td>
</tr>
<tr>
<td>CAMERA</td>
</tr>
<tr>
<td>CENTER</td>
</tr>
<tr>
<td>CHRNUM</td>
</tr>
<tr>
<td>CLPLIN</td>
</tr>
<tr>
<td>CLPPOL</td>
</tr>
<tr>
<td>CLPPEZ</td>
</tr>
<tr>
<td>CNTBAR</td>
</tr>
<tr>
<td>CNTPRT</td>
</tr>
<tr>
<td>COLERR</td>
</tr>
<tr>
<td>COLORS</td>
</tr>
<tr>
<td>COPNUM</td>
</tr>
<tr>
<td>DEFAULT</td>
</tr>
<tr>
<td>DEFINE</td>
</tr>
<tr>
<td>DELETE</td>
</tr>
<tr>
<td>DISALL</td>
</tr>
<tr>
<td>DISTXT</td>
</tr>
<tr>
<td>DODITH</td>
</tr>
<tr>
<td>DRWABS</td>
</tr>
<tr>
<td>DRWBBOR</td>
</tr>
<tr>
<td>ERASE</td>
</tr>
<tr>
<td>EXTCOP</td>
</tr>
<tr>
<td>FNDFRM</td>
</tr>
<tr>
<td>FOURVW</td>
</tr>
<tr>
<td>FRINGE</td>
</tr>
<tr>
<td>FRNBAR</td>
</tr>
<tr>
<td>GENTXT</td>
</tr>
<tr>
<td>GETAGR</td>
</tr>
<tr>
<td>GETCLP</td>
</tr>
<tr>
<td>GETCOM</td>
</tr>
<tr>
<td>GETCOP</td>
</tr>
<tr>
<td>GETDEF</td>
</tr>
<tr>
<td>GETGRU</td>
</tr>
<tr>
<td>GETITM</td>
</tr>
<tr>
<td>GETLIM</td>
</tr>
<tr>
<td>GETNUM</td>
</tr>
<tr>
<td>GETWND</td>
</tr>
</tbody>
</table>
GETXT ........................................ 2.47
GLBACT ....................................... 2.48
GLBAXS ....................................... 2.49
GRAINT ....................................... 2.50
GTUROX ....................................... 2.51

GTUROY ....................................... 2.52
GTUROZ ....................................... 2.53
GTUSCL ....................................... 2.54
GTUTRN ....................................... 2.55
HIDLIN ....................................... 2.56

HIDSUR ....................................... 2.57
HUDITM ....................................... 2.58
INCONG ....................................... 2.59
INITGI ....................................... 2.60
INQBGC ....................................... 2.61

INQBOX ....................................... 2.62
INQCNT ....................................... 2.63
INQCON ....................................... 2.64
INQDEV ....................................... 2.65
INQDSP ....................................... 2.66

INQDST ....................................... 2.67
INQFLG ....................................... 2.68
INQFNC ....................................... 2.69
INQFOG ....................................... 2.70
INQFRM ....................................... 2.71

INQFRN ....................................... 2.72
INQGLB ....................................... 2.73
INQLAF ....................................... 2.74
INQLEV ....................................... 2.75
INQLIT ....................................... 2.76

INQPER ....................................... 2.77
INQPRT ....................................... 2.78
INQRAT ....................................... 2.79
INQREF ....................................... 2.80
INQSCL ....................................... 2.81

INQSPL ....................................... 2.82
INQSUB ....................................... 2.83
INQTEN ....................................... 2.84
INQTXT ....................................... 2.85
INQUIR ....................................... 2.86

INRATG ....................................... 2.87
INSPLG ....................................... 2.88
INT4X4 ....................................... 2.89
INTENG ....................................... 2.90
INTHD2 ....................................... 2.91
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTHID</td>
<td>2.92</td>
</tr>
<tr>
<td>INTLR</td>
<td>2.93</td>
</tr>
<tr>
<td>INTSHA</td>
<td>2.94</td>
</tr>
<tr>
<td>ITABLE</td>
<td>2.95</td>
</tr>
<tr>
<td>LCUC</td>
<td>2.96</td>
</tr>
<tr>
<td>LENTXT</td>
<td>2.97</td>
</tr>
<tr>
<td>LITSRC</td>
<td>2.98</td>
</tr>
<tr>
<td>LODCOL</td>
<td>2.99</td>
</tr>
<tr>
<td>MAPWV</td>
<td>2.100</td>
</tr>
<tr>
<td>MAXMIN</td>
<td>2.101</td>
</tr>
<tr>
<td>MODIFY</td>
<td>2.102</td>
</tr>
<tr>
<td>MOV4X4</td>
<td>2.104</td>
</tr>
<tr>
<td>MOVABS</td>
<td>2.105</td>
</tr>
<tr>
<td>MUL4X4</td>
<td>2.106</td>
</tr>
<tr>
<td>NEWGRP</td>
<td>2.107</td>
</tr>
<tr>
<td>NORMAL</td>
<td>2.108</td>
</tr>
<tr>
<td>NORMTM</td>
<td>2.109</td>
</tr>
<tr>
<td>NORVEC</td>
<td>2.110</td>
</tr>
<tr>
<td>NRMAVE</td>
<td>2.111</td>
</tr>
<tr>
<td>OPNFIL</td>
<td>2.112</td>
</tr>
<tr>
<td>OUTBOX</td>
<td>2.113</td>
</tr>
<tr>
<td>PCHECK</td>
<td>2.114</td>
</tr>
<tr>
<td>PERDST</td>
<td>2.115</td>
</tr>
<tr>
<td>PIPLIN</td>
<td>2.116</td>
</tr>
<tr>
<td>PIPSRC</td>
<td>2.118</td>
</tr>
<tr>
<td>POLFIL</td>
<td>2.120</td>
</tr>
<tr>
<td>POLSTA</td>
<td>2.121</td>
</tr>
<tr>
<td>POLYS</td>
<td>2.122</td>
</tr>
<tr>
<td>PORMAN</td>
<td>2.123</td>
</tr>
<tr>
<td>PROMPT</td>
<td>2.124</td>
</tr>
<tr>
<td>PRSPC</td>
<td>2.125</td>
</tr>
<tr>
<td>PRSSTF</td>
<td>2.126</td>
</tr>
<tr>
<td>PURGE</td>
<td>2.127</td>
</tr>
<tr>
<td>PVEC</td>
<td>2.128</td>
</tr>
<tr>
<td>RDINIT</td>
<td>2.129</td>
</tr>
<tr>
<td>READMV</td>
<td>2.130</td>
</tr>
<tr>
<td>REDCHK</td>
<td>2.131</td>
</tr>
<tr>
<td>RESET</td>
<td>2.132</td>
</tr>
<tr>
<td>ROTATE</td>
<td>2.133</td>
</tr>
<tr>
<td>RPLCOP</td>
<td>2.134</td>
</tr>
<tr>
<td>RPLGLOB</td>
<td>2.135</td>
</tr>
<tr>
<td>RUNANI</td>
<td>2.136</td>
</tr>
<tr>
<td>SAVTXT</td>
<td>2.137</td>
</tr>
<tr>
<td>SCALE</td>
<td>2.138</td>
</tr>
<tr>
<td>SETBGC</td>
<td>2.139</td>
</tr>
</tbody>
</table>

**ALPHABETIZED LIST**
<table>
<thead>
<tr>
<th>Function</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETBOX</td>
<td>2.140</td>
</tr>
<tr>
<td>SETCLP</td>
<td>2.141</td>
</tr>
<tr>
<td>SETCNT</td>
<td>2.142</td>
</tr>
<tr>
<td>SETCOL</td>
<td>2.143</td>
</tr>
<tr>
<td>SETCON</td>
<td>2.144</td>
</tr>
<tr>
<td>SETDEF</td>
<td>2.145</td>
</tr>
<tr>
<td>SETDSP</td>
<td>2.146</td>
</tr>
<tr>
<td>SETDST</td>
<td>2.147</td>
</tr>
<tr>
<td>SETFLG</td>
<td>2.148</td>
</tr>
<tr>
<td>SETFNC</td>
<td>2.149</td>
</tr>
<tr>
<td>SETFOG</td>
<td>2.150</td>
</tr>
<tr>
<td>SETFRN</td>
<td>2.151</td>
</tr>
<tr>
<td>SETGLB</td>
<td>2.152</td>
</tr>
<tr>
<td>SETLIN</td>
<td>2.153</td>
</tr>
<tr>
<td>SETLUT</td>
<td>2.154</td>
</tr>
<tr>
<td>SETMOD</td>
<td>2.155</td>
</tr>
<tr>
<td>SETPER</td>
<td>2.156</td>
</tr>
<tr>
<td>SETPRT</td>
<td>2.157</td>
</tr>
<tr>
<td>SETRAT</td>
<td>2.158</td>
</tr>
<tr>
<td>SETREF</td>
<td>2.159</td>
</tr>
<tr>
<td>SETSCL</td>
<td>2.160</td>
</tr>
<tr>
<td>SETSCN</td>
<td>2.161</td>
</tr>
<tr>
<td>SETSPL</td>
<td>2.162</td>
</tr>
<tr>
<td>SETSTD</td>
<td>2.163</td>
</tr>
<tr>
<td>SETSUB</td>
<td>2.164</td>
</tr>
<tr>
<td>SETTEN</td>
<td>2.165</td>
</tr>
<tr>
<td>SETWND</td>
<td>2.166</td>
</tr>
<tr>
<td>SHRINK</td>
<td>2.167</td>
</tr>
<tr>
<td>SMOCLR</td>
<td>2.168</td>
</tr>
<tr>
<td>SMOGET</td>
<td>2.169</td>
</tr>
<tr>
<td>SMOSTR</td>
<td>2.170</td>
</tr>
<tr>
<td>STCONG</td>
<td>2.171</td>
</tr>
<tr>
<td>STOREL</td>
<td>2.172</td>
</tr>
<tr>
<td>STRATG</td>
<td>2.173</td>
</tr>
<tr>
<td>STSPLG</td>
<td>2.174</td>
</tr>
<tr>
<td>STTENG</td>
<td>2.175</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>2.176</td>
</tr>
<tr>
<td>TEXT</td>
<td>2.177</td>
</tr>
<tr>
<td>TMPTS</td>
<td>2.179</td>
</tr>
<tr>
<td>TMPTSN</td>
<td>2.180</td>
</tr>
<tr>
<td>TRAVRS</td>
<td>2.181</td>
</tr>
<tr>
<td>TXTDIS</td>
<td>2.183</td>
</tr>
<tr>
<td>TXTINT</td>
<td>2.184</td>
</tr>
<tr>
<td>WARPOL</td>
<td>2.185</td>
</tr>
</tbody>
</table>
WRITMV ..................................... 2.186
WRTCHK ..................................... 2.187
WRTDRV ..................................... 2.188
WRTGEO ..................................... 2.189
ZPLSTA ..................................... 2.190
<table>
<thead>
<tr>
<th>ROUTINES AND ITEMS ROUTINES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTGRP</td>
<td>2.2</td>
</tr>
<tr>
<td>ADDCAL</td>
<td>2.3</td>
</tr>
<tr>
<td>COPNUM</td>
<td>2.21</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>2.22</td>
</tr>
<tr>
<td>DEFINE</td>
<td>2.23</td>
</tr>
<tr>
<td>DELETE</td>
<td>2.24</td>
</tr>
<tr>
<td>EXTCOP</td>
<td>2.31</td>
</tr>
<tr>
<td>GETAGR</td>
<td>2.37</td>
</tr>
<tr>
<td>GETCOP</td>
<td>2.40</td>
</tr>
<tr>
<td>GETDEF</td>
<td>2.41</td>
</tr>
<tr>
<td>GETGRU</td>
<td>2.42</td>
</tr>
<tr>
<td>GETITM</td>
<td>2.43</td>
</tr>
<tr>
<td>HUDITM</td>
<td>2.58</td>
</tr>
<tr>
<td>INITGI</td>
<td>2.60</td>
</tr>
<tr>
<td>INQUIR</td>
<td>2.86</td>
</tr>
<tr>
<td>MODIFY</td>
<td>2.102</td>
</tr>
<tr>
<td>NEWGRP</td>
<td>2.107</td>
</tr>
<tr>
<td>PURGE</td>
<td>2.127</td>
</tr>
<tr>
<td>RPLCOP</td>
<td>2.135</td>
</tr>
<tr>
<td>SUMMRY</td>
<td>2.176</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ANIMATION ROUTINES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ANFILI</td>
<td>2.4</td>
</tr>
<tr>
<td>ANFILS</td>
<td>2.5</td>
</tr>
<tr>
<td>ANIDRV</td>
<td>2.6</td>
</tr>
<tr>
<td>BEGANI</td>
<td>2.10</td>
</tr>
<tr>
<td>CAMERA</td>
<td>2.11</td>
</tr>
<tr>
<td>FNDFRM</td>
<td>2.32</td>
</tr>
<tr>
<td>GLBACT</td>
<td>2.48</td>
</tr>
<tr>
<td>INCONG</td>
<td>2.59</td>
</tr>
<tr>
<td>INQCON</td>
<td>2.64</td>
</tr>
<tr>
<td>INQFRM</td>
<td>2.71</td>
</tr>
<tr>
<td>INQAT</td>
<td>2.79</td>
</tr>
<tr>
<td>INQSPL</td>
<td>2.82</td>
</tr>
<tr>
<td>INQSUB</td>
<td>2.83</td>
</tr>
<tr>
<td>INQTEN</td>
<td>2.84</td>
</tr>
<tr>
<td>INRATG</td>
<td>2.87</td>
</tr>
<tr>
<td>INSPLG</td>
<td>2.88</td>
</tr>
</tbody>
</table>
ANIMATION Routines (cont.)

<table>
<thead>
<tr>
<th>Routines</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>INTENG</td>
<td>2.90</td>
</tr>
<tr>
<td>RUNANI</td>
<td>2.136</td>
</tr>
<tr>
<td>SETCON</td>
<td>2.144</td>
</tr>
<tr>
<td>SETRAT</td>
<td>2.158</td>
</tr>
<tr>
<td>SETSPL</td>
<td>2.162</td>
</tr>
<tr>
<td>SETSUB</td>
<td>2.164</td>
</tr>
<tr>
<td>SETTEN</td>
<td>2.165</td>
</tr>
<tr>
<td>STCONG</td>
<td>2.171</td>
</tr>
<tr>
<td>STRATG</td>
<td>2.173</td>
</tr>
<tr>
<td>STSPLG</td>
<td>2.175</td>
</tr>
<tr>
<td>STTENG</td>
<td>2.176</td>
</tr>
</tbody>
</table>

DISPLAY DEVICE Routines

<table>
<thead>
<tr>
<th>Routines</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COLERR</td>
<td>2.19</td>
</tr>
<tr>
<td>DISTXT</td>
<td>2.26</td>
</tr>
<tr>
<td>DRWABS</td>
<td>2.28</td>
</tr>
<tr>
<td>ERASE</td>
<td>2.30</td>
</tr>
<tr>
<td>INQDEV</td>
<td>2.65</td>
</tr>
<tr>
<td>ITABLE</td>
<td>2.95</td>
</tr>
<tr>
<td>MOVABS</td>
<td>2.105</td>
</tr>
<tr>
<td>POLFIL</td>
<td>2.120</td>
</tr>
<tr>
<td>RDINIT</td>
<td>2.129</td>
</tr>
<tr>
<td>SETCOL</td>
<td>2.143</td>
</tr>
<tr>
<td>SETLN</td>
<td>2.153</td>
</tr>
<tr>
<td>SETLUT</td>
<td>2.154</td>
</tr>
<tr>
<td>SETMOD</td>
<td>2.155</td>
</tr>
<tr>
<td>SETSCN</td>
<td>2.161</td>
</tr>
</tbody>
</table>

TRANSFORMATION Routines

<table>
<thead>
<tr>
<th>Routines</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ATFROM</td>
<td>2.8</td>
</tr>
<tr>
<td>AUTOON</td>
<td>2.9</td>
</tr>
<tr>
<td>CENTER</td>
<td>2.12</td>
</tr>
<tr>
<td>FOURVW</td>
<td>2.33</td>
</tr>
<tr>
<td>GETWND</td>
<td>2.46</td>
</tr>
<tr>
<td>GTOUROX</td>
<td>2.52</td>
</tr>
<tr>
<td>GTOUROY</td>
<td>2.54</td>
</tr>
<tr>
<td>GTOUROZ</td>
<td>2.55</td>
</tr>
<tr>
<td>GTUSCL</td>
<td>2.56</td>
</tr>
<tr>
<td>GTUTRN</td>
<td>2.57</td>
</tr>
<tr>
<td>INQDST</td>
<td>2.68</td>
</tr>
<tr>
<td>INQGLB</td>
<td>2.73</td>
</tr>
</tbody>
</table>

LIST BY FUNCTION ix
<table>
<thead>
<tr>
<th>Transformation Routines (cont.)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>INQLAF</td>
<td>2.74</td>
</tr>
<tr>
<td>INQPER</td>
<td>2.77</td>
</tr>
<tr>
<td>INQPRT</td>
<td>2.78</td>
</tr>
<tr>
<td>INQSCL</td>
<td>2.81</td>
</tr>
<tr>
<td>INT4X4</td>
<td>2.89</td>
</tr>
<tr>
<td>MAPWV</td>
<td>2.100</td>
</tr>
<tr>
<td>MOV4X4</td>
<td>2.104</td>
</tr>
<tr>
<td>MUL4X4</td>
<td>2.106</td>
</tr>
<tr>
<td>PERDST</td>
<td>2.115</td>
</tr>
<tr>
<td>PORMAN</td>
<td>2.123</td>
</tr>
<tr>
<td>PRSPC</td>
<td>2.125</td>
</tr>
<tr>
<td>PRSSTF</td>
<td>2.126</td>
</tr>
<tr>
<td>RESET</td>
<td>2.132</td>
</tr>
<tr>
<td>ROTATE</td>
<td>2.133</td>
</tr>
<tr>
<td>RPLGLB</td>
<td>2.135</td>
</tr>
<tr>
<td>SCALE</td>
<td>2.138</td>
</tr>
<tr>
<td>SETDST</td>
<td>2.147</td>
</tr>
<tr>
<td>SETGLB</td>
<td>2.152</td>
</tr>
<tr>
<td>SETPER</td>
<td>2.156</td>
</tr>
<tr>
<td>SETPRT</td>
<td>2.157</td>
</tr>
<tr>
<td>SETSCL</td>
<td>2.160</td>
</tr>
<tr>
<td>SETWND</td>
<td>2.161</td>
</tr>
<tr>
<td>SHRINK</td>
<td>2.162</td>
</tr>
<tr>
<td>TMPTS</td>
<td>2.179</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text Routines</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CHRNUM</td>
<td>2.13</td>
</tr>
<tr>
<td>GENTXT</td>
<td>2.36</td>
</tr>
<tr>
<td>INQTXT</td>
<td>2.85</td>
</tr>
<tr>
<td>SAVTXT</td>
<td>2.137</td>
</tr>
<tr>
<td>TEXT</td>
<td>2.177</td>
</tr>
<tr>
<td>TXTDIS</td>
<td>2.183</td>
</tr>
<tr>
<td>TXTINT</td>
<td>2.184</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vector and Scalar Function Routines</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AROCLR</td>
<td>2.7</td>
</tr>
<tr>
<td>CNTBAR</td>
<td>2.17</td>
</tr>
<tr>
<td>FRINGE</td>
<td>2.34</td>
</tr>
<tr>
<td>FRNBAR</td>
<td>2.35</td>
</tr>
</tbody>
</table>

List by Function x
VECTOR AND SCALAR FUNCTION ROUTINES (cont.)

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INQCNT</td>
<td>2.63</td>
</tr>
<tr>
<td>INQDSP</td>
<td>2.66</td>
</tr>
<tr>
<td>INQFNC</td>
<td>2.69</td>
</tr>
<tr>
<td>INQFRN</td>
<td>2.72</td>
</tr>
<tr>
<td>INQLEV</td>
<td>2.75</td>
</tr>
<tr>
<td>INQREF</td>
<td>2.80</td>
</tr>
<tr>
<td>PVEC</td>
<td>2.128</td>
</tr>
<tr>
<td>SETCNT</td>
<td>2.142</td>
</tr>
<tr>
<td>SETDSP</td>
<td>2.146</td>
</tr>
<tr>
<td>SETFNC</td>
<td>2.149</td>
</tr>
<tr>
<td>SETFRN</td>
<td>2.151</td>
</tr>
<tr>
<td>SETREF</td>
<td>2.159</td>
</tr>
<tr>
<td>SETSTD</td>
<td>2.163</td>
</tr>
</tbody>
</table>

INPUT/OUTPUT ROUTINES

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETCOM</td>
<td>2.39</td>
</tr>
<tr>
<td>GETNUM</td>
<td>2.45</td>
</tr>
<tr>
<td>GETTXT</td>
<td>2.47</td>
</tr>
<tr>
<td>LCUC</td>
<td>2.96</td>
</tr>
<tr>
<td>LENTXT</td>
<td>2.97</td>
</tr>
<tr>
<td>OPNFILE</td>
<td>2.112</td>
</tr>
<tr>
<td>PROMPT</td>
<td>2.124</td>
</tr>
</tbody>
</table>

RASTER ROUTINES

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLORS</td>
<td>2.20</td>
</tr>
<tr>
<td>DODITH</td>
<td>2.27</td>
</tr>
<tr>
<td>HIDSUR</td>
<td>2.57</td>
</tr>
<tr>
<td>INQFOG</td>
<td>2.70</td>
</tr>
<tr>
<td>INQLIT</td>
<td>2.76</td>
</tr>
<tr>
<td>INTHD2</td>
<td>2.91</td>
</tr>
<tr>
<td>INTHID</td>
<td>2.92</td>
</tr>
<tr>
<td>INTSHA</td>
<td>2.94</td>
</tr>
<tr>
<td>LITSRC</td>
<td>2.98</td>
</tr>
<tr>
<td>PCHECK</td>
<td>2.114</td>
</tr>
<tr>
<td>PIPSRC</td>
<td>2.118</td>
</tr>
<tr>
<td>SETFOG</td>
<td>2.150</td>
</tr>
<tr>
<td>WARPOL</td>
<td>2.185</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
</tr>
<tr>
<td>HIDLIN</td>
<td>2.56</td>
</tr>
<tr>
<td>INTTHLR</td>
<td>2.93</td>
</tr>
<tr>
<td>LODCOL</td>
<td>2.99</td>
</tr>
<tr>
<td>STOREL</td>
<td>2.172</td>
</tr>
</tbody>
</table>

**Hidden Line Removal Routines**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL</td>
<td>2.108</td>
</tr>
<tr>
<td>NORMTM</td>
<td>2.109</td>
</tr>
<tr>
<td>NORVEC</td>
<td>2.110</td>
</tr>
<tr>
<td>NRMAVE</td>
<td>2.111</td>
</tr>
<tr>
<td>SMOCLR</td>
<td>2.163</td>
</tr>
<tr>
<td>SMOGET</td>
<td>2.169</td>
</tr>
<tr>
<td>SMOOSTR</td>
<td>2.170</td>
</tr>
<tr>
<td>TMPTSN</td>
<td>2.180</td>
</tr>
</tbody>
</table>

**Normal Creation and Manipulation Routines**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLPLIN</td>
<td>2.14</td>
</tr>
<tr>
<td>CLPPOL</td>
<td>2.15</td>
</tr>
<tr>
<td>CLPPZEP</td>
<td>2.16</td>
</tr>
<tr>
<td>GETCLP</td>
<td>2.38</td>
</tr>
<tr>
<td>POLSTA</td>
<td>2.121</td>
</tr>
<tr>
<td>SETCLP</td>
<td>2.141</td>
</tr>
<tr>
<td>ZPLSTA</td>
<td>2.190</td>
</tr>
</tbody>
</table>

**Clipping Routines**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>READMV</td>
<td>2.130</td>
</tr>
<tr>
<td>REDCHK</td>
<td>2.131</td>
</tr>
<tr>
<td>WRTMV</td>
<td>2.186</td>
</tr>
<tr>
<td>WRTCHK</td>
<td>2.187</td>
</tr>
<tr>
<td>WRTDRV</td>
<td>2.189</td>
</tr>
<tr>
<td>WRTGEO</td>
<td>2.190</td>
</tr>
</tbody>
</table>

**Input/Output File Routines**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAINT</td>
<td>2.50</td>
</tr>
</tbody>
</table>

**Initialization Routine (Must be Included in All User Applications)**

**Other Routines**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTRPT</td>
<td>2.18</td>
</tr>
<tr>
<td>DISALL</td>
<td>2.19</td>
</tr>
<tr>
<td>DRWBOR</td>
<td>2.29</td>
</tr>
<tr>
<td>GETLIM</td>
<td>2.44</td>
</tr>
<tr>
<td>GLBAXS</td>
<td>2.49</td>
</tr>
</tbody>
</table>
### OTHER ROUTINES (cont.)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INQBGC</td>
<td>2.61</td>
</tr>
<tr>
<td>INQBOX</td>
<td>2.62</td>
</tr>
<tr>
<td>INQFLG</td>
<td>2.67</td>
</tr>
<tr>
<td>MAXMIN</td>
<td>2.100</td>
</tr>
<tr>
<td>OUTBOX</td>
<td></td>
</tr>
<tr>
<td>PIPLIN</td>
<td>2.113</td>
</tr>
<tr>
<td>POLYS</td>
<td>2.116</td>
</tr>
<tr>
<td>SETBGC</td>
<td>2.122</td>
</tr>
<tr>
<td>SETBOX</td>
<td>2.139</td>
</tr>
<tr>
<td>SETDEF</td>
<td></td>
</tr>
<tr>
<td>SETFLG</td>
<td>2.140</td>
</tr>
<tr>
<td>TRAVRS</td>
<td>2.145</td>
</tr>
<tr>
<td></td>
<td>2.148</td>
</tr>
<tr>
<td></td>
<td>2.181</td>
</tr>
</tbody>
</table>
Introduction

THE RASTER GRAPHICS DISPLAY LIBRARY

Raster Display Systems

When you use computer graphics to see complex three-dimensional data, your accuracy and productivity increase. Raster display systems show trends and patterns much more clearly than volumes of tabulated data because color and shading can be used to depict realistic scenes.

In engineering applications, color and shading may be used not only to realistically portray an object, but to present additional information. Distorted shapes and color coding of functions include failure criteria, stress or strain components, temperature, and pressure.

A Common Problem of Raster Display Systems

Many applications lack adequate presentation graphics. You may not have a set of software tools that allow you to generate quickly the raster image capabilities you need. Or the algorithms you want may be incorporated in commercial packages or proprietary codes, or be unavailable outside a specialized university environment. What you need is a raster display system that allows you to display the graphics you want.

The Raster Graphics Display Library

The Raster Graphics Display Library (RGDL) is a high level subroutine package that gives you the advanced raster graphics display capabilities you need. RGDL uses FORTRAN source code routines to build subroutines modular enough to use as stand-alone routines in a black box type of environment. With RGDL, you have a set of tools that are easily used, well documented, and error tolerant.

How To Use This Manual

This documentation is divided into four chapters and three appendices. The first chapter contains six examples you should go through if you want to learn how to use RGDL in the fastest, most complete way possible. Chapter two contains a technical reference section, chapter three a common block reference section, and chapter four a reference on include files.

The first appendix, a link map, contains information from MOVIE.BYU software and documentation. MOVIE.BYU is a general purpose computer graphics display system that uses RGDL software.

Appendix B is a section about hierarchical data structures. The third appendix contains installation information and our address.
If you want more information about each section, go to the next page or to the chapter introductions. If you want to make sure the contents on the magnetic tape you received are complete, go to page C.1. For installation information and our address, see page C.2.
CHAPTER SUMMARIES

Chapter 1  EXAMPLE PROBLEMS

The display library will be introduced to you using six example problems. When you compile and link these examples, you will get an understanding for what you can achieve using the library routines. Each example builds on the previous problem and gets more complicated from one example to the next.

You won't need to type in the example problems, because they are supplied with the software. Make sure you compile and link applications with the proper system flags set. Applications may become very large and some systems require special compiler and link options.

Chapter 2  TECHNICAL REFERENCES

Chapter three contains routines within the display library that you will call to produce raster graphics. The routines are in alphabetical order, each on a separate page. Each user-callable routine is described by function and calling parameters.

Chapter three does not contain a description of all routines in the display library, but rather only those routines that could be called by an applications program.

Chapter 3  COMMON BLOCK REFERENCES

Chapter four contains a common block reference. The chapter lists all common blocks that are used in the display library, and discusses the use of each variable within each common block.

Chapter 4  INCLUDE FILE REFERENCES

Chapter five contains a reference on the include files that are necessary to compile the display library. Each include file and its purpose are listed.

Appendix A  LINK MAP

This appendix contains the link map for MOVIE.BYU version 6. MOVIE.BYU is a general purpose computer graphics display system that uses RGDl software. It uses the display library in addition to other routines, and is an excellent source as an auxiliary reference.

The link map is included to show you the amount of code that may be necessary to perform a function. The link map will also help if you don't have library utilities.
Appendix B  AN INTRODUCTION TO HIERARCHICAL DATA STRUCTURES

Appendix B contains a section about creating new groups. The section will help you understand the hierarchical data structures used in RGD L. You are asked to create new groups in Examples Five and Six of Chapter One.

Appendix C  SOFTWARE INSTALLATION

Appendix C contains installation information. It also gives you the address and phone number of the people you can contact if you find bugs or problems in RGD L software.
Chapter 1

EXAMPLE PROBLEMS

This chapter contains example drivers for the Raster Graphics Subroutine Library. The examples call routines described in chapter two, Technical References. You will learn how to create applications programs that use the graphics library routines. By going through the examples, you will also become familiar with many of the system's capabilities.

Why should you go through the examples?

Because you will save time and headaches. The examples are not busywork. They are six exercises that build on each other to teach you how the Raster Graphics Subroutine Library works.

You should study the examples carefully, then actually compile, link, and run them. You won't need to type them in because they are supplied on tape with the rest of the software.

Things you should know

Make sure you compile and link applications with the correct system flags set. The applications may become very large and some systems require special compiler and link options.

All examples are read in a hard-coded geometry file, EXP9.GEO. A common data base is used so you can follow along with the example problems through more complex operations. Of course, in most cases, you will want to create drivers that prompt for the geometry file names to be read into the data base.

The examples shown do not perform error checking. We strongly suggest you include error checking in applications you write.

About the display device and host computer

The examples will be most useful to you when they are run on a raster display device that allows for screen overlays, such as the Tektronix 4115.

Because no link maps are shown in the examples, the host computer you use should have a library utility. If the host computer does not have a library utility, refer to the link map in the appendix to see what routines must be linked with the drivers.

After the first example, all new additions to the next five exercises are shown in bold.
EXAMPLE 1

PROGRAM RSPEX1

RASTER GRAPHICS SUBROUTINE PACKAGE EXAMPLE 1

THIS EXAMPLE PROBLEM WILL READ IN A GEOMETRY FILE AND
DISPLAY IT IN LINE DRAWING MODE WITH ALL DEFAULTS.

THIS CALL IS NEEDED IN ALL PROGRAMS. IT INITIALIZES THE
GRAPHICS PACKAGE.

CALL GRAINT

READ IN A MOVIE.BYU GEOMETRY FILE CALLED EXP9.GEO. DO NOT READ IN
A FUNCTION OR A DISPLACEMENT FILE.

CALL READMV('EXP9.GEO',' ',' ')

DISPLAY LINE DRAWING PICTURE

CALL DISALL
STOP
END
EXAMPLE 2

PROGRAM RSPEX2

RASTER GRAPHICS SUBROUTINE PACKAGE EXAMPLE 2

THIS EXAMPLE PROBLEM ADDS A USER DEFINED LOOK FROM VECTOR TO EXAMPLE PROBLEM 1

REAL LOOKAT(3), LOOKFR(3)

THIS SECTION IS NEEDED IN ALL PROGRAMS. IT INITIALIZES THE GRAPHICS PACKAGE.

CALL GRAINT

READ IN A MOVIE.BYU GEOMETRY FILE CALLED EXP9.GEO. DO NOT READ IN A FUNCTION OR A DISPLACEMENT FILE.

CALL READMV('EXP9.GEO',' ',' ') 

PROMPT FOR LOOK FROM VECTOR. SET THE LOOK AT POINT TO ZERO, AND CALL THE ROUTINE TO SET UP THE LOOK AT/FROM TRANSFORMATION.

WRITE(*,*),'ENTER LOOK-FROM VECTOR (X,Y,Z)'
READ(*,*)LOOKFR(1),LOOKFR(2),LOOKFR(3)
LOOKAT(1) = 0.
LOOKAT(2) = 0.
LOOKAT(3) = 0.
CALL ATFROM(LOOKAT,LOOKFR)

COULD ALSO DO THE ATFROM CALL BE GOING THROUGH GLBACT, THIS CALL WOULD LOOK LIKE:

CALL GLBACT('FRM',LOOKAT(1),LOOKAT(2),LOOKAT(3),LOOKFR(1),
& 'LOOKFR(2),LOOKFR(3))

GLBACT SHOULD BE USED IF A USER IS RUNNING ANIMATION, SINCE THIS ROUTINE WILL CAPTURE THE LOOK AT/FROM COMMAND AS A MOVEMENT THAT NEEDS TO BE ANIMATED.

DISPLAY LINE DRAWING OF PICTURE

CALL DISALL
STOP
END
EXAMPLE 3

PROGRAM RSPEX3

RASTER GRAPHICS SUBROUTINE PACKAGE EXAMPLE 3

THIS EXAMPLE PROBLEM USES THE I/O ROUTINES FOR ALL
PROMPTS AND USER INPUT. ALSO ADDED IS A USER DEFINED
ROTATION AND TRANSLATION OF PART 1. THE PICTURE IS
DISPLAYED IN LINE DRAWING MODE.

CHARACTER*(73) TEXT
CHARACTER*80 FORM
DIMENSION XNUM(40), VALUE(6)
REAL LOOKAT(3), LOOKFR(3)

CALL GRAINT

READ IN A MOVIE.BYU GEOMETRY FILE CALLED EXP9.GEO. DO NOT READ IN
A FUNCTION OR A DISPLACEMENT FILE.

CALL READMV('EXP9.GEO', ' ', ' ')

PROMPT FOR LOOK FROM VECTOR. SET THE LOOK AT POINT TO ZERO, AND
CALL THE ROUTINE TO SET UP THE LOOK AT/FROM TRANSFORMATION.
PERFORM AN INTERNAL READ FOR THE PROMPT. CALL GETXT WITH THE
PROMPT, WHICH WILL RETURN THE TEXT STRING RESPONSE FROM THE USER -
TEXT, AND THE NUMBER OF CHARACTERS IN TEXT - NCHAR. NEXT, CALL
GETNUM WHICH WILL PARSE THE USER TEXT STRING FOR NUMBERS. XNUM
WILL BE AN ARRAY OF NUMBERS, AND K2 IS THE NUMBER OF NUMBERS
FOUND.

WRITE(FORM,100)'ENTER LOOK-FROM VECTOR (X,Y,Z)'
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
LOOKAT(1) = 0
LOOKAT(2) = 0
LOOKAT(3) = 0
IF(K2 .EQ 0) THEN
    WRITE(FORM,100)'NO LOOK-FROM VECTOR SPECIFIED!'
    CALL PROMPT(FORM)
ELSE
    LOOKFR(1) = XNUM(1)
    LOOKFR(2) = XNUM(2)
    LOOKFR(3) = XNUM(3)
    CALL ATFROM (LOOKAT,LOOKFR)
END IF
EXAMPLE 3

PROMPT FOR AND PERFORM A ROTATION FOR PART 1

WRITE(FORM,100) 'X,Y,Z ROTATION ANGLES (DEGREES) FOR PART 1?'

CALL GETXT(TEXT,NCHAR,FROM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
VALUE(1) - XNUM(1)
VALUE(2) - XNUM(2)
VALUE(3) - XNUM(3)

'1' IS THE PART TO MODIFY.
4 IS THE FLAG INDICATING ROTATION.
VALUE IS AN ARRAY OF NUMBERS CORRESPONDING TO THE ACTION FLAG.

CALL MODIFY('1',4,VALUE)

PROMPT FOR AND PERFORM A TRANSLATION FOR PART 1.

WRITE(FORM,100) 'X,Y,Z TRANSLATION FOR PART 1'
CALL GETXT(TEXT,NCHAR,FROM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
VALUE(1) - XNUM(1)
VALUE(2) - XNUM(2)
VALUE(3) - XNUM(3)

THE ACTION FLAG 5 IS THE TRANSLATION FLAG

CALL MODIFY('1',5,VALUE)

DISPLAY LINE DRAWING OF PICTURE

CALL DISALL

FORMAT(A)
STOP
END
EXAMPLE 4

PROGRAM RSPEX4

RASTER GRAPHICS SUBROUTINE PACKAGE EXAMPLE 4

THIS EXAMPLE PROBLEM TAKES EXAMPLE PROBLEM 3, AND DEFINES FOUR
DIFFERENT VIEW PORTS. THE FIRST VIEW PORT IS DRAWN IN LINE
DRAWING MODE, THE SECOND VIEW PORT IS DRAWN IN HIDDEN LINE MODE,
THE THIRD VIEW PORT IS DRAWN IN SHADED IMAGE MODE. THE FOURTH
VIEW PORT IS DRAWN IN SHADED IMAGE MODE WITH DITHERING.

CHARACTER*(73) TEXT
CHARACTER*80 FORM
DIMENSION XNUM(40),VALUE(6)
REAL LOOKAT(3), LOOKFR(3)

This section is needed in all programs. It initializes the
graphics package.

CALL GRAINT

READ IN A MOVIE.BYU GEOMETRY FILE CALLED EXP9.GEO. DO NOT READ IN
A FUNCTION OR A DISPLACEMENT FILE.

CALL READMV('EXP9.GEO', ' ', ' ')

PROMPT FOR LOOK FROM VECTOR. SET THE LOOK AT POINT TO ZERO, AND
CALL THE ROUTINE TO SET UP THE LOOK AT/FROM TRANSFORMATION.
PERFORM AN INTERNAL READ FOR THE PROMPT. CALL GETXT WITH THE
PROMPT, WHICH WILL RETURN THE TEXT STRING RESPONSE FROM THE USER-
TEST, AND THE NUMBER OF CHARACTERS IN TEXT-NCHAR. NEXT, CALL
GETNUM WHICH WILL PARSE THE USER TEXT STRING FOR NUMBERS. XNUM
WILL BE AN ARRAY OF NUMBERS, AND K2 IS THE NUMBER OF NUMBERS
FOUND.

WRITE(FORM,100) 'ENTER LOOK-FROM VECTOR (X,Y,Z)'
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
LOOKAT(1) = 0.
LOOKAT(2) = 0.
LOOKAT(3) = 0.
IF(K2 .EQ. 0) THEN
WRITE(FORM,100) 'NO LOOK-FROM VECTOR SPECIFIED!'
CALL PROMPT(FORM)
ELSE
LOOKFR(1) = XNUM(1)
LOOKFR(2) = XNUM(2)
LOOKFR(3) = XNUM(3)
CALL ATFROM(LOOKAT,LOOKFR)
END IF
EXAMPLE 4

PROMPT FOR AND PERFORM A ROTATION FOR PART 1

WRITE(FORM,100) 'X,Y,Z ROTATION ANGLES (DEGREES) FOR PART 1?'
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
VALUE(1) = XNUM(1)
VALUE(2) = XNUM(2)
VALUE(3) = XNUM(3)
CALL MODIFY('1',4,VALUE)

PROMPT FOR AND PERFORM A TRANSLATE FOR PART 1

WRITE(FORM,100) 'X,Y,Z TRANSLATION FOR PART 1?'
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
VALUE(1) = XNUM(1)
VALUE(2) = XNUM(2)
VALUE(3) = XNUM(3)
CALL MODIFY('1',5,VALUE)

SET THE VIEWPORT TO THE UPPER LEFT QUARTER OF THE SCREEN
CALL SETPRT(0.,.5,.75)
DISPLAY LINE DRAWING OF PICTURE
CALL DISALL

SET THE VIEWPORT TO THE UPPER RIGHT QUARTER OF THE SCREEN
CALL SETPRT(.5,1.,.75)

SET THE HIDDEN LINE FLAG ON AND DISPLAY THE PICTURE
CALL SETFLG('HIDDEN', .TRUE.)
CALL DISALL

SET THE VIEWPORT TO THE LOWER LEFT QUARTER OF THE SCREEN
CALL SETPRT(0.,.5,.25)
SET THE SHADED IMAGE FLAG AND DISPLAY THE PICTURE
CALL SETFLG('SHADED', .TRUE.)
CALL DISALL

SET THE VIEWPORT TO THE LOWER RIGHT QUARTER OF THE SCREEN
CALL SETPRT(.5,1.,.25)

SET THE DITHERING FLAG ON, AND DISPLAY THE RASTER IMAGE OVER
CALL SETFLG('DITHER', .TRUE.)
CALL DISALL

C

100 FORMAT(A)
STOP
END
EXAMPLE 5

PROGRAM RSPEX5

RASTER GRAPHICS SUBROUTINE PACKAGE EXAMPLE 5

THIS EXAMPLE PROBLEM TAKES EXAMPLE PROBLEM 4 AND MODIFIES IT
BY CREATING A NEW GROUP. THE COLOR AND SHADING OF PART 1 IS SET,
AND THE COLOR OF THE HIGHER GROUP IS SET TO GREEN. THE UPPER
RIGHT VIEWPORT IS SHOWN IN FOUR VIEW MODE.

CHARACTER*(73) TEXT
CHARACTER*80 FORM
DIMENSION XNUM(40), VALUE(6)
REAL LOOKAT(3), LOOKFR(3)

CALL GRAINT

READ IN A MOVIE.BYU GEOMETRY FILE CALLED EXP9.GEO. DO NOT READ IN
A FUNCTION OR A DISPLACEMENT FILE.

CALL READMV('EXP9.GEO',' ',' ','

PROMPT FOR LOOK FROM VECTOR. SET THE LOOK AT POINT TO ZERO, AND
CALL THE ROUTINE TO SET UP THE LOOK AT/FROM TRANSFORMATION.
PERFORM AND INTERNAL READ FOR THE PROMPT. CALL GETXT WITH THE
PROMPT, WHICH WILL RETURN THE TEXT STRING RESPONSE FROM THE USER -
TEST, AND THE NUMBER OF CHARACTERS IN TEXT - NCHAR. NEXT, CALL
GETNUM WHICH WILL PARSE THE USER TEXT STRING FOR NUMBERS. XNUM
WILL BE AN ARRAY OF NUMBERS, AND K2 IS THE NUMBER OF NUMBERS
FOUND.

WRITE(FORM,100) 'ENTER LOOK-FROM VECTOR (X,Y,Z)'
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
LOOKAT(1) = 0.
LOOKAT(2) = 0.
LOOKAT(3) = 0.
IF(K2 .EQ. 0) THEN
  WRITE(FORM,100) 'NO LOOK-FROM VECTOR SPECIFIED!'
  CALL PROMPT(FORM)
ELSE
  LOOKFR(1) = XNUM(1)
  LOOKFR(2) = XNUM(2)
  LOOKFR(3) = XNUM(3)
  CALL ATFROM(LOOKAT,LOOKFR)
END IF
EXAMPLE 5

PROMPT FOR AND PERFORM A ROTATION FOR PART 1

WRITE(FORM, 100) 'X,Y,Z ROTATION ANGLES (DEGREES) FOR PART 1?'

CALL GETXT(TEXT,NCHAR,FMT)
CALL GETXT(TEXT,NCHAR,XNUM,K2)
VALUE(1) - XNUM(1)
VALUE(2) - XNUM(2)
VALUE(3) - XNUM(3)

CALL MODIFY('1',4,VALUE)

PROMPT FOR AND PERFORM A TRANSLATE FOR PART 1

WRITE(FORM,100) 'X,Y,Z TRANSLATION FOR PART 1?'
CALL GETXT(TEXT,NCHAR,FMT)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
VALUE(1) - XNUM(1)
VALUE(2) - XNUM(2)
VALUE(3) - XNUM(3)
CALL MODIFY('1',5,VALUE)

CREATE A NEW GROUP (SEE "AN INTRODUCTION TO HIERARCHICAL DATA STRUCTURES" IN APPENDIX B.)

CALL DEFINE('TOP GROUP')

COPY THE GROUP "ROOT" INTO THE NEW GROUP AND CALL THE COPY "COPY ROOT 1"

CALL ADDCAL('ROOT','COPY ROOT 1')

SET THE COLOR OF THE NEW COPY TO GREEN, AND SET THE COLOR OF PART 1 (PART 1 IS REALLY COPY 1) TO RED. ALSO SET THE SHADING ATTRIBUTE OF COPY 1 TO SMOOTH

VALUE(1) - 0.
VALUE(2) - 1.
VALUE(3) - 0.
CALL MODIFY('COPY ROOT 1',8,VALUE)
VALUE(1) - 1.
VALUE(2) - 0.
CALL MODIFY('1',8,VALUE)
VALUE(1) - 2.
CALL MODIFY('1',11,VALUE)

SET THE VIEWPORT TO THE UPPER LEFT QUARTER OF THE SCREEN

CALL SETPRT(0.,.5,.75)

DISPLAY LINE DRAWING OF PICTURE
CALL DISALL

SET THE VIEWPORT TO THE UPPER RIGHT QUARTER OF THE SCREEN

CALL SETPRT(.5,1.,.75)

SET THE HIDDEN LINE FLAG ON AND DISPLAY THE PICTURE IN FOUR
STANDARD VIEWS

CALL SETFLG('HIDDEN', .TRUE.)
CALL FOURVW

SET THE VIEWPORT TO THE LOWER LEFT QUARTER OF THE SCREEN

CALL SETPRT(0.,.5,.25)

SET THE SHADED IMAGE FLAG AND DISPLAY THE PICTURE

CALL SETFLG('SHADED', .TRUE.)
CALL DISALL

SET THE VIEWPORT TO THE LOWER RIGHT QUARTER OF THE SCREEN

CALL SETPRT(.5,1.,.25)

SET THE DITHERING FLAG ON, AND DISPLAY THE RASTER IMAGE OVER

CALL SETFLG('DITHER', .TRUE.)
CALL DISALL

100 FORMAT(A)
STOP
END
EXAMPLE 6

PROGRAM RSPEX6

RASTER GRAPHICS SUBROUTINE PACKAGE EXAMPLE 6

THIS EXAMPLE PROBLEM TAKES EXAMPLE PROBLEM 5 AND MODIFIES IT
BY CREATING TWO COPIES OF THE ORIGINAL GEOMETRY FILE. THE USER
IS PROMPTED FOR A TRANSLATE OF THE SECOND COPY.

CHARACTER*(73) TEXT
CHARACTER*80 FORM
DIMENSION XNUM(40),VALUE(6)
REAL LOOKAT(3), LOOKFOR(3)

C THIS SECTION IS NEEDED IN ALL PROGRAMS. IT INITIALIZES THE
GRAPHICS PACKAGE.
C
CALL GRAINT
C
READ IN A MOVIE.BYU GEOMETRY FILE CALLED EXP9.GEO. DO NOT READ IN
A FUNCTION OR A DISPLACEMENT FILE.

CALL READMV('EXP9.GEO',' ',' ')

PROMPT FOR LOOK FROM VECTOR. SET THE LOOK AT POINT TO ZERO, AND
CALL THE ROUTINE TO SET UP THE LOOK AT/FROM TRANSFORMATION.
PERFORM AND INTERNAL READ FOR THE PROMPT. CALL GETXT WITH THE
PROMPT, WHICH WILL RETURN THE TEXT STRING RESPONSE FROM THE USER -
TEST, AND THE NUMBER OF CHARACTERS IN TEXT - NCHAR. NEXT, CALL
GETNUM WHICH WILL PARSE THE USER TEXT STRING FOR NUMBERS. XNUM
WILL BE AN ARRAY OF NUMBERS, AND K2 IS THE NUMBER OF NUMBERS
FOUND.

WRITE(FORM,100) 'ENTER LOOK-FROM VECTOR (X,Y,Z)'
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
LOOKAT(1) = 0.
LOOKAT(2) = 0.
LOOKAT(3) = 0.
IF(K2 .EQ. 0) THEN
    WRITE(FORM,100) 'NO LOOK-FROM VECTOR SPECIFIED!'
    CALL PROMPT(FORM)
ELSE
    LOOKFR(1) = XNUM(1)
    LOOKFR(2) = XNUM(2)
    LOOKFR(3) = XNUM(3)
    CALL ATFROM(LOOKAT,LOOKFR)
END IF

PROMPT FOR AND PERFORM A ROTATION FOR PART 1
EXAMPLE 6

WRITE(FORM,100) 'X,Y,Z ROTATION ANGLES (DEGREES) FOR PART 1?'

CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
LOOKAT(1) = (1) = XNUM(1)
LOOKAT(2) = (2) = XNUM(2)
LOOKAT(3) = (3) = XNUM(3)
CALL MODIFY('1',4,VALUE)

PROMPT FOR AND PERFORM A TRANSLATE FOR PART 1

WRITE(FORM,100) 'X,Y,Z TRANSLATION FOR PART 1?'  
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
VALUE(1) = XNUM(1)
VALUE(2) = XNUM(2)
VALUE(3) = XNUM(3)
CALL MODIFY('1',5,VALUE)

CREATE A NEW GROUP (SEE "AN INTRODUCTION TO HIERARCHICAL DATA STRUCTURES" IN APPENDIX B.)

CALL DEFINE('TOP GROUP')

COPY THE GROUP "ROOT" INTO THE NEW GROUP AND CALL THE COPY "COPY ROOT 1"

CALL ADDCAL('ROOT','COPY ROOT 1')

COPY ANOTHER COPY OF ROOT INTO THE NEW GROUP AND CALL THE COPY "COPY ROOT 2"

CALL ADDCAL('ROOT','COPY ROOT 2')

PROMPT THE USER FOR A TRANSLATION OF "COPY ROOT 2"

WRITE(FORM,100) 'X,Y,Z TRANSLATION FOR COPY ROOT 2?'  
CALL GETXT(TEXT,NCHAR,FORM)
CALL GETNUM(TEXT,NCHAR,XNUM,K2)
VALUE(1) = XNUM(1)
VALUE(2) = XNUM(2)
VALUE(3) = XNUM(3)
CALL MODIFY('COPY ROOT 2',5,VALUE)

SET THE COLOR OF "COPY ROOT 1" TO GREEN, AND SET THE COLOR OF PART 1(PART 1 IS REALLY COPY 1) TO RED. ALSO SET THE SHADING ATTRIBUTE OF COPY 1 TO SMOOTH

VALUE(1) = 0.
VALUE(2) = 1.
EXAMPLE 6

VALUE(3) = 0.
CALL MODIFY('COPY ROOT 1', 8, VALUE)
VALUE(1) = 1.
VALUE(2) = 0.
CALL MODIFY('1', 8, VALUE)
VALUE(1) = 2.
CALL MODIFY ('1', 11, VALUE)

SET THE VIEWPORT TO THE UPPER LEFT QUARTER OF THE SCREEN

CALL SETPRT(0., .5, .75)

DISPLAY LINE DRAWING OF PICTURE

CALL DISALL

SET THE VIEWPORT TO THE UPPER RIGHT QUARTER OF THE SCREEN

CALL SETPRT(.5, 1., .75)

SET THE HIDDEN LINE FLAG ON AND DISPLAY THE PICTURE IN FOUR STANDARD VIEWS

CALL SETFLG('HIDDEN', .TRUE.)
CALL FOURVW

SET THE VIEWPORT TO THE LOWER LEFT QUARTER OF THE SCREEN

CALL SETPRT(0., .5, .25)

SET THE SHADED IMAGE FLAG AND DISPLAY THE PICTURE

CALL SETFLG('SHADED', .TRUE.)
CALL DISALL

SET THE VIEWPORT TO THE LOWER RIGHT QUARTER OF THE SCREEN

CALL SETPRT(.5, 1., .25)

SET THE DITHERING FLAG ON, AND DISPLAY THE RASTER IMAGE OVER

CALL SETFLG('DITHER', .TRUE.)
CALL DISALL

100 FORMAT(A)
STOP
END
Chapter Two

TECHNICAL REFERENCES

This chapter contains detailed descriptions of all the user-callable subroutines within RGDL. Use the information to write application programs for your needs.

The routines are listed in alphabetical order, each on a separate page. The function is listed at the top of each page, together with the name of the routine. The function of each routine is given, as well as parameters, common blocks, include blocks, and other routines that are used in conjunction with the given routine.

It is unlikely you will use the routines in alphabetical order, so we have provided an index of the routines according to their function on page vi, after the table of contents. An alphabetized index is also given on page iii. The index divided into functions will probably be the easiest way to find an unknown routine with a known function.
DECLARATION: Subroutine ACTGRP (grunam)

FUNCTION: Sets the active group to an existing group.

PARAMETERS

in:      grunam     The name of an existing group to be set active.
character: grunam(*)
DECLARATION: Subroutine ADDCAL (giname, cpynam)

FUNCTION: Adds a call (item or group) to the active group.

PARAMETERS

in:  
giname Name of group or item to be added to the active group.

cpynam Copy name that giname will be referred to.

character: giname*, cpynam*

COMMON BLOCKS: /CHFWAS/
/CLIMIT/
/CMEMRY/
DECLARATION: Entry ANFILI (inqnam)

FUNCTION: Returns the animation extension.

An animation file extension of '@#$' indicates that animation is not open.

PARAMETERS

out: inqnam Name of the animation extension.

class: inqnam(3)

SEE ALSO: ANFILS
DECLARATION:  Entry ANFILS (newnam)

FUNCTION:  Stores the animation extension.

An animation file extension of '@#$' indicates that animation is not open. Use ANIDRV to open animation.

PARAMETERS

in:  newnam  Name of the new animation extension.

class:  newnam(3)

SEE ALSO:  ANFILI

ANIDRV
DECLARATION: Subroutine ANIDRV

FUNCTION: Drives the animation module. ANIDRV does many basic functions like opening and closing the animation module, setting up subframes, and setting up spline control.

All animation interfacing should be performed through ANIDRV.

COMMON BLOCKS: /CHFWAS/
/DEVI/
/CMEMRY/
DECLARATION: Entry AROCLR

FUNCTION: Clears the displacement arrow arrays.

Call AROCLR once for each part containing displacement vector arrows.
DECLARATION: Subroutine ATFROM (lookat, lookfr)

FUNCTION: Creates the viewing matrix based on the look-from and look-at positions.

Coordinates are in world coordinates.

PARAMETERS

in: lookat Look-at position.
    lookfr Look-from position.

real: lookat(3), lookfr(3)

COMMON BLOCKS: /VEWSTF/

SEE ALSO: INQLAF
DECLARATION: Subroutine AUTOON

FUNCTION: Turns auto center off.

COMMON BLOCKS: /DEV/ /VEWSTF/
DECLARATION: Subroutine BEGANI (aninam)

FUNCTION: Starts the animation, opens necessary files, and initializes all appropriate variables.

PARAMETERS

in: aninam The three character extension names used for animation files.

character: aninam(3)

COMMON BLOCKS: /CHFWAS/
/CLIMIT/
/CMEMRY/
/DEVI/
DECLARATION: Subroutine CAMERA

FUNCTION: Acts as the interface to a camera for animation.

CAMERA gets called after each frame is displayed. You can modify this routine for anything you wish to do after a frame is shown. For example, you may want to interface to a computer-controlled camera.
 DECLARATION: Subroutine CENTER

FUNCTION: Sets up the viewing matrix, if not already specified. Also sets up the window and clipping planes to center the model.

The limits of the scene to be displayed must first be set up in SETLIM. Normally, this is done by a call to TRAVRS.

COMMON BLOCKS: /DEVI/
/VEWSTF/
/RESOLT/

SEE ALSO: SETLIM
GETLIM
TRAVRS
Debeclaration: Subroutine CHRNUM (number, lenchr, chrint, nchars)

Function: Converts integer data into an equivalent character string.

Negative numbers are not allowed.

Parameters

In: lenchr  Maximum length of the string.
      number  The integer number to be made into a character string.

Out: chrint  The character string representation of an integer.
        nchars  Maximum number of characters in the string to be returned.

Integer: nchars, lenchr, number

Character: chrint(*)
DECLARATION: Subroutine CLPLIN (corbeg, corend, wxmin, wxmax, wymin, wymax, plot)

FUNCTION: Clips lines in x and y.

All coordinates are eye coordinates.

PARAMETERS

in: corbeg X,Y,Z coordinate triplet of beginning point; may be modified after clipping.
corend X,Y,Z ending triplet of line may be modified after clipping.
wxmin Minimum X window coordinate.
wxmax Maximum X window coordinate.
wymin Minimum Y window coordinate.
wymax Maximum Y window coordinate.

out: plot .TRUE. if line is visible or partly visible.

real: corbeg(3), corend(3), wxmin, wxmax, wymin, wymax

logical: plot

SEE ALSO: CLPPOL
          CLPPZE
DECLARATION: Subroutine CLPPOL (nedges, cordin, polnin, funcin, clpfun, clpnor, wxmin, wxmax, wymin, wymax, plot)

FUNCTION: CLPOLL clips polygons against the x and y boundaries.

The coordinate, normal, and function arrays must close the polygon. For example, the first vertex location is the same as the nedges + 1 location.

All coordinates are eye coordinates.

PARAMETERS

in: clpfun .TRUE. if clipping function values are being clipped.
clpnor .TRUE. if clipping normals are being clipped.
wxmin Minimum x window coordinate.
wxmax Maximum x window coordinate.
wymin Minimum y window coordinate.
wymax Maximum y window coordinate.
nedges Number of edges in the polygon.
cordin Coordinates of polygon.
cordin(1,1) = X coordinate of first polygon vertex
cordin(2,1) = Y coordinate of first polygon vertex
Cordin may be modified after clipping.

polnin Normals of polygon after clipping.
polinin(1,1) = X component of unit normal at vertex 1
polnin(2,1) = Y component of unit normal at vertex 1
Polnin may be modified after clipping.

funcin Function values of polygon after clipping.
Funcin may be modified after clipping.

out: plot Logical; .TRUE. if polygon is visible after clipping.

real: wxmin, wxmax, wymin, wymax, cordin(3,*), polnin(3,*),
funcin(*)

integer: nedges

logical: clpfun, clpnor, plot

SEE ALSO: CLPPZE
CLPLIN
DECLARATION: Subroutine CLPPZE (nedges, cordin, polnin, funcin, clpfun, clpnor, zmin, zmax, plot)

FUNCTION: Clips polygons against the z boundaries.

PARAMETERS

in: clpfun .TRUE. if clipping function values are being clipped.
clpnor .TRUE. if clipping normals are being clipped.
zmin Distance from eye to nearest clipping plane.
zmax Distance from eye to farthest clipping plane.
nedges Number of edges in the polygon. Nedges may be modified after clipping.
cordin Coordinates of polygon. Cordin may be modified after clipping.
polnin Normals of polygon. Polnin may be modified after clipping.
funcin Function values of polygon. Funcin will be modified after clipping.

out: plot .TRUE. if polygon is visible.

real: zmin, zmax, cordin (3,*), polnin (3,*), funcin (*)

integer: nedges

logical: plot, clpfun, clpnor

SEE ALSO: CLPPOL
          CLPLIN

Chapter Two  TECHNICAL REFERENCES  2.16
DECLARATION: Subroutine CNTBAR

FUNCTION: Outputs contour legend bar for contours.

Contours should be set up CNTBAR before calling this routine by calling SETFRN.

SEE ALSO: SETFLG
INQFRN
SETFRN
INQFLG
SETCNT
DECLARATION: Subroutine CNTRPT (nedges, coords, xcentr)

FUNCTION: Calculates a polygon's center point.

The coordinate array should close the polygon.

PARAMETERS

in:  nedges  Number of vertices in a polygon.
     coords  Coordinate array for polygon.

out: xcentr  Coordinate array of center point.

real: coords(3,*), xcentr(3)

integer: nedges
DECLARATION: Function COLERR (color)

FUNCTION: Calculates the error between the desired color and the next lowest color in the look-up table.

PARAMETERS

- **in:** color  The intensity of either the r, g, or b component of the desired color. (The range of color is from 0.0 to 1.0).
- **real:** color
DECLARATION: Subroutine COLORS (nedges, atribt, polfun, coords, polnor, colnod, trnnod)

FUNCTION: Computes color values at polygon nodes according to user-defined light sources.

All coordinates are in the eye coordinate system.

PARAMETERS

in: nedges Number of vertices in the polygon.
     atribt Attribute list for polygon.
     coords Coordinate array for polygon vertices.
     polfun Polygon function values at the vertices.
     polnor Normals of the polygon vertices.

out: colnod Red, green, and blue color intensity at polygon vertices.
     trnnod Transparency values of polygon vertices.

real: atribt(*), colnod(3,*), coords(3,*), polfun(*), trnnod(*), polnor(3,*)

integer: nedges

COMMON BLOCKS: /CFLAGS/
/CATTRI/
/CHFWAS/
/CMEMRY/
DECLARATION: Subroutine COPNUM (sordex, numcop, jroot, jchild, itorgr, cpynam)

FUNCTION: Given the copy number, gets the copy name together with its copy item and parent.

PARAMETERS

   out: sordex  Sort number associated with the instance.
          cpynam  Copy name associated with sordex.
          jroot   Root group number.
          jchild  Item or group number of the copy name.
          itorgr  Flag 1 if a group; 0 if an item.
          numcop  Copy number.

   integer: sordex, jroot, jchild, itorgr, numcop

   character: cpynam(*)

SEE ALSO: GETCOP
DECLARATION: Subroutine DEFALT (atlist)

FUNCTION: Sets up a default attributes list.

PARAMETERS

out: atlist Current attribute list.
real: atlist (maxiaa)

COMMON BLOCKS: /CATTRI/
/CLIMIT/
DECLARATION: SubroutineDEFINE (grunam)

FUNCTION: Defines and activate a new group.

PARAMETERS

in: grunam The name of the new group to be activated.
character: grunam(*)
DECLARATION: Subroutine DELETE (cpynam)

FUNCTION: Deletes a call to an instance.

PARAMETERS

in: cpynam Copy name to be deleted from the database.

character: cpynam(*)
DECLARATION: Subroutine DISALL

FUNCTION: Displays the picture for all options.

PARAMETERS

COMMON BLOCKS: /DEV1/ /CMEMRY/ /CHFWAS/ /CLIMIT/
**DECLARATION:** Subroutine DISTXT (string, nchars, r, g, b, ix, iy)

**FUNCTION:** Displays text strings.

**PARAMETERS**

- **in:**
  - string: Text string to be displayed.
  - nchars: Number of characters in string.
  - r, g, b: Desired color of text (0.0-1.0).
  - ix, iy: Origin of the string in screen coordinates.

- **real:** r, g, b

- **integer:** nchars, ix, iy

- **character:** string*(*)
DECLARATION: Subroutine DODITH (r,g,b)

FUNCTION: DODITH performs dithering for a pixel.

PARAMETERS

in:  r,g,b  Color components of the pixel (0.0 - 1.0) modified on return.

real: r,g,b
**DECLARATION:** Subroutine DRWABS (ix,iy)

**FUNCTION:** Draws a line from the current location to the specified point (ix,iy).

**PARAMETERS**

in: ix,iy The screen coordinates of the point to be drawn to. The (0,0) point is in the lower left corner, x increases from left to right, and y increases from bottom to top. The ix,iy point should become the current point after the draw command.

integer: ix,iy
DECLARATION: Subroutine DRWBOR

FUNCTION: Draws a border around the current viewport.

When using DRWBOR, you must put graphics device into and out of the graphics mode.

SEE ALSO: SETMOD
DECLARATION: Subroutine ERASE

FUNCTION: Clears the current viewport to the background color.
DECLARATION: Subroutine EXTCOP (cpynam, ctm, atlist)

FUNCTION: Extracts transformation and attribute information for an instance.

PARAMETERS

in: cpynam  The instance name.
out: atlist  Attribute list for the instance.
      ctm    Transformation matrix for the instance.
real: atlist(maxatt), ctm(4,4)

character: cpynam(*)

COMMON BLOCKS: /CHFWAS/
                /CLIMIT/
                /CMEMRY/

SEE ALSO: RPLCOP
DECLARATION: Subroutine FNDFRM (chrfrm, keyfrm, subfrm, ierr)

FUNCTION: Finds the keyframe and subframe from a character string of the form keyframe.subframe.

PARAMETERS

in: chrfrm A character string defined in the form keyfrm.subfrm.

out: keyfrm Keyframe number.
subfrm Subframe number.
ierr Error flag. 0 if operation was successful; 1 if chrfrm was not in the form keyframe.subframe.

integer: subfrm, ierr, keyframe

character: chrfrm*(*)

Chapter Two TECHNICAL REFERENCES 2.32
DECLARATION: Subroutine FOURVW

FUNCTION: FOURVW displays the four standard views.

COMMON BLOCKS: /CFLAGS/
/RESOLT/
**DECLARATION:** Subroutine FRINGE (nlevel, colfnc, fnclcv, valnod, xir, xig, xib)

**FUNCTION:** Calculates color at a node of a polygon when fringes are on.

*Fringes should be set up using SETFRN before calling FRINGE.*

**PARAMETERS**

- **in:**
  - colfnc: Color at each fringe function level.
  - fnclcv: Function value at each level.
  - nlevel: Number of function levels.
  - valnod: Function value at the node.

- **out:**
  - xir: Red color component at node.
  - xig: Green color component at node.
  - xib: Blue color component at node.

- **real:**
  - colfnc(3,nlevel), fnclcv(nlevel), xir, xig, xib, valnod

- **integer:** nlevel

**SEE ALSO:** SETFLG

FRNBAR

INQFRN

SETFRN

INQFLG
DECLARATION: Subroutine FRNBAR

FUNCTION: Displays fringe bar (legend) for color fringes.

*Fringes should be set using SETFRN before calling FRNBAR.*

SEE ALSO: FRINGE
SETFLG
INQFRN
SETFRN
INQFLG
DECLARATION: Entry GENTXT

FUNCTION: Generates text string for display.

SEE ALSO: TXTDIS
          TXTINT
DECLARATION: Entry GETAGR (i)

FUNCTION: Returns the active group number.

PARAMETERS

out: i Active group number.

integer: i

SEE ALSO: NEWGRP
GETCLP

DECLARATION: Entry GETCLP (znear, zfar, zdist)

FUNCTION: Gets z clipping information.

PARAMETERS

out: znear Distance from eye to the nearest z clipping plane.
     zfar Distance from eye to the farthest z clipping plane.
     zdist Distance from znear to zfar.

real: znear, zfar, zdist

SEE ALSO: SETCLP
DECLARATION: Subroutine GETCOM (text, nchar, word, n, key, kl)

FUNCTION: Parses a text string for key words.

PARAMETERS

in: n Number of words in the key word array.
    word Array of key words that are being looked for.
    nchar Number of characters in entered text string.
    text Entered text string to be parsed.

out: kl Number of key words found in the text string.
    key Array of key word starting locations in the word array that were found in the text string.

integer: n, kl, nchar, key(*)
character: text*75, word(n)*4

SEE ALSO: GETNUM
DECLARATION: Subroutine GETCOP (cpynam, iflag, jroot, jchild, itorgr, sordex, numcop, postfl)

FUNCTION: Given the copy name, gets the copy number and associated indexes.

PARAMETERS

in: cpynam Copy name to get information on.
    iflag 1 if adding a copy name.
    2 if deleting a copy name.
    3 if traversing with the copy name to get group and item information.
    4 if purging all calls to an item or group.
    5 if printing the copy names to the screen.

out: postfl 0 if the copy is posted; 1 if unposted.
    sordex Sort number for groups and items.
    itorgr Flag 1 if a group; 0 if an item.
    numcop Copy number associated with the copy name.
    jroot Root group number.
    jchild Item or group number of the copy name.

real: postfl

integer: iflag, sordex, itorgr, numcop

character: cpynam(*)

SEE ALSO: COPNUM
DECLARATION: Entry GETDEF (numcol, red, grn, blu)

FUNCTION: Returns the red, green, and blue color intensities for color number numcol.

All color values are from 0 to 1.

PARAMETERS

out: numcol Number of the default color to return information for.
red Red intensity default color number numcol.
green Green intensity default color number numcol.
blu Blue intensity default color number numcol.

real: red, grn, blu

integer: numcol

SEE ALSO: SETDEF
DECLARATION: Subroutine GETGRU (grpnam, jroot, iflag)

FUNCTION: Gets group name and group number relationships.

PARAMETERS

in: grpnam Name of the group. Grpnam will be in or out depending on iflag option).
    iflag 1 if adding a group name to the list.
            2 if deleting a group from the list.
            3 if getting the name from the number.
            4 if getting the number from the name.
            5 if listing out defined group names.
            6 if changing the name of a group.

out: jroot Number of the group. Jroot will be in or out depending on iflag option).
    integer: jroot, iflag
    character: grpnam(*)

SEE ALSO: GETITM
DECLARATION: Subroutine GETITM (itname, jchild, iflag)

FUNCTION: Gets item name and item number relationships.

PARAMETERS

in: itname Name of the item. Itname will be in or out depending on iflag option.
   iflag 1 if adding a item name to the list.
          2 if deleting a item from the list.
          3 if getting the name from the number.
          4 if getting the number from the name.
          5 if listing out defined item names.

out: jchild Number of the item (in or out depending on iflag option).

integer: jchild, iflag

character: itname(*)

SEE ALSO: GETGRU
GETLIM

DECLARATION: Entry GETLIM (small, big)

FUNCTION: Returns the maximum and minimum model coordinates of all parts processed with MAXMIN.

PARAMETERS

out: big Maximum coordinates returned.
    small Minimum coordinates returned.

real: small(3), big(3)

COMMON BLOCKS: /CLIMIT/  
                 /MASTER/  

SEE ALSO: MAXMIN
DECLARATION: Subroutine GETNUM (text, nchar, xnum, k2)

FUNCTION: Parses a text string for numbers.

PARAMETERS

in:  nchar   Number of characters in line of entered text.
     text     Entered text string to be parsed.

out: k2       Number of numbers found in the text string.
           xnum     Array of numbers found in the text string.

real: xnum(40)

integer: k2, nchar

character: text*73

SEE ALSO: GETCOM
DECLARATION:  Entry GETWND (xmin, xmax, ymin, ymax)

FUNCTION:  Inquires on the world window bounds. The scene
is clipped to these values.

PARAMETERS

out:  xmin  Minimum x world coordinate for window currently set.
     xmax  Maximum x world coordinate for window currently set.
     ymin  Minimum y world coordinate for window currently set.
     ymax  Maximum y world coordinate for window currently set.

real:  xmin, xmax, ymin, ymax

SEE ALSO:  SETWND
**DECLARATION:** Subroutine GETXT (text, nchar, textin)

**FUNCTION:** Prompts for and retrieves a text string.

**PARAMETERS**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>in:</td>
<td>textin Optional prompt string that will be written to display device prior to reading text string. If textin is blank, the prompt will not be displayed.</td>
</tr>
<tr>
<td>out:</td>
<td>nchar Number of characters in line of entered text.</td>
</tr>
<tr>
<td></td>
<td>text Entered text string.</td>
</tr>
<tr>
<td>integer:</td>
<td>nchar</td>
</tr>
<tr>
<td>character:</td>
<td>text<em>73, textin</em>80</td>
</tr>
</tbody>
</table>
DECLARATION: Subroutine GLBACT (code, val1, val2, val3, val4, val5, val6)

FUNCTION: Provides an interface between global actions and the animate module.

It is not necessary to call GLBACT unless animation is being done.

PARAMETERS

in: code Code that determines the operation performed in GLBACT.

'SCL' Global scales for this subroutine.
'ROT' Global rotate.
'COL' Background color.
'WND' Window.
'PER' Perspective angle.
'DST' Distance.
'FRM' Look-from position or vector.
'ATT' Look-for position.
'CLP' Z-clipping planes.
'VEW' Viewport.
'LIT' Lightsourcing information.
'FOG' Fog planes.

val1 Val1 through val6 are values needed to perform wanted operation.

real: val1, val2, . . . val6

character: code*3

COMMON BLOCKS: /CMEMRY/
/CHFWAS/
DECLARATION: Subroutine GLBAXS (tm, atribt)

FUNCTION: Draws the global axis.

PARAMETERS

in: tm Viewing matrix.
    atribt Attribute list.

real: tm(4,4), atribt (maxiaa)

COMMON BLOCKS: /CATTRI/
                /CFLAGS/
                /CLIMIT/
                /RESOLT/
                /VEWSTF/
DECLARATION: Subroutine GRAINT

FUNCTION: Initializes the graphics package.

GRAINT must always be called before any other graphics application is called.

COMMON BLOCKS: /CATTRI/
/CHFWAS/
/CLIMIT/
/CMEMRY/
/DEVI/
/VEWSTF/
DECLARATION: Subroutine GTUROX (theta, ctm)

FUNCTION: Concatenates a right-handed rotation about the x-axis to a transformation matrix.

PARAMETERS

in: theta Angle in degrees.
ctm Current transformation matrix that is modified on return.

real: theta, ctm(4,4)

SEE ALSO: GTUROY
          GTUROZ
          GTUSCL
          GTUTRN
          INT4X4
DECLARATION: Subroutine GTUROY (theta, ctm)

FUNCTION: Concatenates a right-handed rotation about y-axis to a transformation matrix.

PARAMETERS

in: theta Angle in degrees.
     ctm Current transformation matrix modified on return.

real: theta, ctm(4,4)

SEE ALSO: GTUROX
          GTUROZ
          GTUSCL
          GTUTRN
          INT4X4
DECLARATION: Subroutine GTUROZ (theta, ctm)

FUNCTION: Concatenates a right-handed rotation about z-axis to a transformation matrix.

PARAMETERS

in: theta Angle in degrees.
    ctm Current transformation matrix modified on return.

real: theta, ctm(4,4)

SEE ALSO: GTUROX
          GTUROY
          GTUSCL
          GTUTRN
          INT4X4
DECLARATION: Subroutine GTUSCL (sx, sy, sz, ctm)

FUNCTION: Concatenates scalings to a transformation matrix.

PARAMETERS

in: sx Scaling in x-direction.
    sy Scaling in y-direction.
    sz Scaling in z-direction.
    ctm Current transformation matrix modified on return.

real: sx, sy, sz, ctm(4,4)

SEE ALSO: GTUROX
          GTUROY
          GTUROZ
          GTUTRN
          INT4X4
DECLARATION: Subroutine GTUTRN (dx, dy, dz, ctm)

FUNCTION: Concatenates a translation to a transformation matrix.

PARAMETERS

in:  dx  Translation in x-direction.
     dy  Translation in y-direction.
     dz  Translation in z-direction.
     ctm Current transformation matrix modified on return.

real: dx, dy, dz, ctm(4,4)

SEE ALSO: GTUROX
          GTUROY
          GTUROZ
          GTUSCL
          INT4X4
DECLARATION: Subroutine HIDLIN (prtcol)

FUNCTION: Perform hidden line removal and contour generation.

HIDLIN can be called after all polygons have been loaded with STOREL.

PARAMETERS

in: prtcol Array of rgb colors for each part in the picture.
real: prtcol(3,*)

COMMON BLOCKS: /HIDSTF/
/KEEP/

SEE ALSO: STOREL
DECLARATION: Subroutine HIDSUR

FUNCTION: Performs hidden surface removal.

HIDSUR can be called after calls to INTHID and INTHD2, and after all polygons have been packed down.

COMMON BLOCKS: /RESOLT/ /CMEMRY/ /CHFWAS/

SEE ALSO: INTHID INTHD2
DECLARATION: Subroutine HUDITM (jpar, jitm, jcopy)

FUNCTION: Deletes an item call from the hierarchical data structure.

PARAMETERS:

in: jpar   Parent group number.
      jcopy  Instance copy number.
      jitm   Item number.

integer: jpar, jcopy, jitm
DECLARATION:  Entry INCONG (congin)

FUNCTION:  Returns the continuity for the look-at and look-from locations for animation.

PARAMETERS

out: congin  Array of continuity values.
real: congin(maxprt,2)
DECLARATION: Subroutine INITGI (grpdex, grsort, trndex, gstart, sgrcnt, sitcnt, namgrp, namitm, igrupa, ichida, iorga, numcpy, namcpy)

FUNCTION: Initializes groups and items.

PARAMETERS

in: grpdex Group number array.
grsort Group sort array.
igrupa Root group array.
trndex Transformation number index array.
gstart Group start array.
sgrcnt Subgroup counter array.
sitcnt Subitem counter array.
namgrp Group names array.
namitm Item names array.
ichida Child array.
iorga Item or group flag array.
numcpy Copy number array.
namcpy Copy name array.

integer: namgrp(maxchr,0:maxgrp)
namitm(maxchr,0:maxitm)
namcpy(maxchr,0:maxins)
igrupa(maxins)
ichida(maxins)
iorga(maxins)
umcpy(maxins)
grpdex(maxins)
grsort(maxins)
trndex(maxins)
gstart(maxgrp)
sgrcnt(maxgrp)
sitcnt(maxgrp)
**DECLARATION:** Entry INQBGC (red, green, blue)

**FUNCTION:** Returns on the background color.

*All values are from 0 to 1.*

**PARAMETERS**

- `out`: red  Red intensity of background color.
  green  Green intensity of background color.
  blue  Blue intensity of background color.

- `real`: red, green, blue

**SEE ALSO:** SETBGC
INQBOX

DECLARATION: Entry INQBOX (npart, cormax, cormin)

FUNCTION: Returns the x, y, z limits of a bounding box for an instance.

PARAMETERS

in: npart Number of parts in the model.
    cormax Maximum x, y, and z coordinates of box.
    cormin Minimum x, y, and z coordinates of box.

real: cormax(3), cormin(3)

integer: npart

SEE ALSO: SETBOX
DECLARATION: Entry INQ_CNT (iflag)

FUNCTION: Returns the number of defined contour levels.

PARAMETERS

out: iflag    Number of contour levels to be displayed.

integer: iflag

SEE ALSO: SETCNT
DECLARATION: Entry INQCON (coninq)

FUNCTION: Returns the instance continuities at the keyframes for animation.

PARAMETERS

out: coninq Array of continuities.
real: coninq(maxfrm,maxprt)
DECLARATION: Subroutine INQDEV (shaded, polfil, hrdwar)

FUNCTION: Checks to see if the device has special hardware capabilities that are being taken advantage of in display. INQDEV also checks to see if the device can fill polygons or do shading.

PARAMETERS

out: hrdwar .TRUE. if special hardware available.
polfil .TRUE. if polygon fills available.
shaded .TRUE. if shaded images possible.

logical: hrdwar, polfil, shaded
DECLARATION: Entry INQDSP \( (x\text{dis}, \ y\text{dis}, \ z\text{dis}, \ \text{valnod}) \)

FUNCTION: Returns the magnitude of a displacement at a node.

PARAMETERS

in: \( x\text{dis} \) X displacement of a node.
\( y\text{dis} \) Y displacement of a node.
\( z\text{dis} \) Z displacement of a node.

out: \( \text{valnod} \) Magnitude of displacement of the node.

real: \( x\text{dis}, \ y\text{dis}, \ z\text{dis}, \ \text{valnod} \)
Entry INQDST (dist)

Inquires distance from eye to look-at point.

Use INQDST only in perspective angle mode and when not doing auto center.

PARAMETERS

in: dist Distance from eye to look-at point.
real: dist

COMMON BLOCKS: /VEWSTF/

SEE ALSO: SETDST
DECLARATION: Entry INQFLG (cflag, lflag)

FUNCTION: Inquires on global flags set.

PARAMETERS

in: cflag Key that indicates what is being inquired on one of the following:

'ALIA' Anti-aliasing flag.
'SHDO' Shadow flag.
'AXIS' Axis are to be drawn flag.
'BOUN' Draw bounding box instead of model flag.
'BORD' Draw border flag.
'CENT' Auto center flag.
'COLO' Draw colored lines flag.
'DISP' Displacements are performed flag.
'DITH' Dithering status flag.
'ELEM' Element labeling flag.
'FOUR' Four view flag.
'GLAS' Transparency flag.
'Haze' Haze/fog flag.
'HIDD' Hidden line flag.
'LAGE' Label contours flag.
'LEGE' Display contour legend flag.
'NODE' Node labeling flag.
'ONEP' Warped polygons on edge flag.
'OVER' Overlay flag.
'PEPA' Perspective or windowing toggle flag.
'PERS' Perspective flag.
'POLY' Polygon clipping flag.
'POOR' Poorman flag.
'SHAD' Shaded images flag.
'SYMM' Symmetric contours or fringes flag.
'VECT' Vector arrows flag.
'WARP' Function warping flag.
'TEXT' Text display flag.
'TK41' Tektronix 4129 hardware flag.
'LAVY' Label for views flag.
'PLYF' Polygon fill shaded images flag.

out: lflag .TRUE. or .FALSE. status of cflag.

integer: iflag

character: cflag(*)

COMMON BLOCKS:/CFLAGS/

SEE ALSO: SETFLG
DECLARATION: Entry INQFNC (nlevel, valmin, valmax)

FUNCTION: Returns the number of color fringes set, and the minimum and maximum range of fringes.

SETFNC should be called before calling INGFNC.

PARAMETERS

out: nlevel   Number of function levels defined.
valmin   Minimum value of color fringes.
valmax   Maximum value of color fringes.

real: valmin, valmax

integer: nlevel

SEE ALSO: SETFNC
INQFOG

DECLARATION: Entry INQFOG (dstn, dstf)

FUNCTION: Returns the distance to the nearest and farthest fog planes.

PARAMETERS

out: dstn Distance to nearest fog plane.
    dstf Distance to farthest fog plane.

real: dstn, dstf

SEE ALSO: SETFOG
**INQFRM**

**DECLARATION:** Entry INQFRM (numfrm)

**FUNCTION:** Returns on the animation keyframe being worked on.

You should call ANFILI before calling INQFRM to see if animation is really going on.

**PARAMETERS**

- **out:** numfrm  The number of the keyframe being worked on.
- **integer:** numfrm

**SEE ALSO:** SETFRM
ANFILI
DECLARATION: Entry INQFRN (iflag)

FUNCTION: Returns the number of fringe levels.

PARAMETERS

out: iflag  Number of fringe levels to be displayed.
integer: iflag

SEE ALSO: SETFRN
DECLARATION: Entry INQGLB (tm)

FUNCTION: Returns the global transformation matrix.

PARAMETERS

out: tm Transformation matrix.
real: tm(4,4)

SEE ALSO: SETGLB
          ROTATE
          SCALE
          RESET
INQLAF

DECLARATION: Subroutine INQLAF (lookat, lookfr, icentr)

FUNCTION: Returns the look-at and look-from position. If icentr is .TRUE., INQLAF will return a vector for lookfr.

PARAMETERS

in: icentr .TRUE. if auto-center is on.
out: lookat Look-at position.
     lookfr Look-from position.
     real: lookat(3), lookfr(3)
     logical: icentr

COMMON BLOCKS: /VEWSTF/
INQLEV

DECLARATION: Entry INQLEV (nlevel, colfnc, fnclen)

FUNCTION: Inquires on the number of function levels, and the function
value and color used with each level for fringes.

PARAMETERS

out: nlevel  Number of function levels defined.
colfnc  Array of colors at each function level.
fnclen  Array of function values at each level.

real: colfnc (3,maxlev), fnclen (3,maxlev)

integer: nlevel
DECLARATION: Subroutine INQLIT (litnum, xltint, litloc, xltxyz)

FUNCTION: Given the light source number, INQLIT returns intensity of the light, whether or not it is at the eye of the observer, and returns its position if not at the eye of the observer.

PARAMETERS

in:  litnum Maximum number of lights allowed.

out: xltint Light source intensity.
     litloc Light source location flag:
             0 = at the eye of the observer
             1 = at a specified location
             2 = on a specified vector at infinity
     xltxyz x,y,z light source locations.

real: xltint, xltxyz (3)

integer: litnum, litloc

COMMON BLOCKS: /CHFWAS/
                /CMEMRY/

SEE ALSO: LITSRC
DECLARATION: Entry INQPER (anginq)

FUNCTION: Inquires perspective angle.

PARAMETERS

out: anginq  Perspective angle that is currently set (in degrees).
real: anginq

SEE ALSO: SETPER
          INQFLG
INQPRT

DECLARATION: Entry INQPRT (ileft, iright, ibottm, itop)

FUNCTION: Returns the viewport boundaries.

All coordinates are screen coordinates.

PARAMETERS

out:  ileft  Left viewport boundary.
      iright Right viewport boundary.
      ibottm Bottom viewport boundary.
      itop  Top viewport boundary.

integer: ileft, iright, itop, ibottm

SEE ALSO: SETPRT
DECLARATION: Entry INQRAT (speed)

FUNCTION: Returns the instance rates at the keyframes for animation.

PARAMETERS

out: speed  Array of rates.
real: speed(maxfrm,maxprt)
INQREF

DECLARATION: Entry INQREF (ncolor, colfnc)

FUNCTION: Inquires on the reference colors for color fringes.

PARAMETERS

out: ncolor Number of reference colors defined.
colfnc Array of colors at each function level.
real: colfnc (3, maxlev)
integer: ncolor

SEE ALSO: INQLEV
INQFNC
SETREF
INQSCL

DETECTION: Entry INQSCL (scalex, scaley)

FUNCTION: Inquires on the window to viewport scale factors.

SETSCL must be called before calling this routine.

PARAMETERS

out: scalex  X direction scale factor.
    scaley  Y direction scale factor.

real: scalex, scaley

COMMON BLOCKS: /RESOLT/

SEE ALSO: SETSCL
DECLARATION: Entry INQSPL (splinq)

FUNCTION: Returns the instance on spline flags at the keyframes for animation.

PARAMETERS

out: splinq Array of spline flags for global.
real: splinq(maxfrm,maxprt)
DECLARATION: Entry INQSUB (isub)

FUNCTION: Returns the number of subframes that exist between the frames for animation.

PARAMETERS

out: isub Array of subframes.

integer: isub(maxfrm)

SEE ALSO: INQFRM ANFILI
DECLARATION: Entry INQTEN (teninq)

FUNCTION: Returns the instance tensions at the keyframes for animation.

PARAMETERS

out: teninq Array of tension for global.
real: teninq(maxfrm, maxprt)
**DECLARATION:** Entry INQTXT (mesage, xsize, ysize, twodim, xpos, ypos, zpos, xang, yang, zang, r, g, b, xsllant, i, nsav)

**FUNCTION:** Inquires on the text string attributes.

**PARAMETERS**

in:  
- i  Index of text string to retrieve.

out:  
- nsav Total number of stored messages.
- mesage The text string array.
- xsize X dimension factor.
- ysize Y dimension factor.
- twodim If .TRUE. message is 2-D.
  If .FALSE., strings are in 3-D text.
- xpos X position of text string in world coordinate.
- ypos Y position of text string in world coordinate.
- zpos Z position of text string in world coordinate.
- xang X directional vector.
- yang Y directional vector.
- zang Z directional vector. Angle of text string if 2-d.
- r, g, b Red, green and blue color values.
- xsllant Rotation about x-axis for 3-d text string in degrees.

real:  
- xsize, ysize, twodim, xpos, ypos, xang, yang, zang, r, g, b, xsllant

integer:  
- i, nsav

character:  
- *80 mesage

**SEE ALSO:** SAVTXT
DECLARATION: Subroutine INQUIR (name)

FUNCTION: Inquires and prints attribute list for an item or group.

PARAMETERS

in: name Copy name.
character: name*(*)
INRATG

**DECLARATION:** Entry INRATG (ratgin)

**FUNCTION:** Returns the array of rates for global animation parameters.

**PARAMETERS**

- out: ratgin  Array of rates for global values.
- real: ratgin(maxfrm,11+litnum)
**DECLARATION:**  Entry INSPLG (onsgin)

**FUNCTION:**  Inquire on spline flag array for the look-at and look-from locations for animation.

**PARAMETERS**

- **out:**  onsgin     Array of on spline flags for instances.
- **real:**  onsgin(maxfrm,2)
DECLARATION: Subroutine INT4X4 (trnfm)

FUNCTION: Initializes a transformation matrix to the identity matrix.

PARAMETERS

out: trnfm Transformation matrix initialized to the identity matrix.

real: trnfm(4,4)
DECLARATION: Entry INTENG (tengin)

FUNCTION: Returns the tension for the look-at and look-from locations for animation.

PARAMETERS

out: tengin  Tension array.
real: tengin(maxprt,2)
DECLARATION: Subroutine INTHD2

FUNCTION: Initializes portions of hidden surface processor.

\textit{INTHID2 should only be run before each hidden surface picture after the call to POLYS.}

COMMON BLOCKS: \texttt{/HIDSTF/}

SEE ALSO: INTHID
DELLARATION: Subroutine INTHID

FUNCTION: Initializes portions of hidden surface processor.

This routine should be run once for each hidden surface picture, before the call to polys.

COMMON BLOCKS: /CHFWAS/
/CMEMRY/
/HIDSTF/
/PIXSTF/
/KEEP/

SEE ALSO: INTHD2
**DECLARATION:** Subroutine INTHLR

**FUNCTION:** Initializes the hidden line removal process.

INTHLR must be called before any other hidden line removal routine.

**COMMON BLOCKS:**

/HIDSTF/
/KEEP/
**DECLARATION:** Subroutine INTSHA (numlit, drknes)

**FUNCTION:** Stores light number and shadow darkness for a light source.

Shadow darkness is linear from 0 to 1.

**PARAMETERS**

- **in:** numlit Number of light sources being set.
- drknes Shadow intensity between 0 and 1. A value of 0 is no shadows, a small value is a dark shadow, and a value close to 1.0 is a very light shadow.
- **real:** drknes
- **integer:** numlit

**COMMON BLOCKS:** /CHFWAS/
/CLIMIT/
/CMEMRY/
ITABLE

DECLARATION: Function ITABLE (r,g,b, errfac)

FUNCTION: Returns the look-up table location of the color nearest the specified color.

PARAMETERS

ian: r,g,b Component intensities of the color looked for (0.0 - 1.0).

errfac Equal to 0.0 for dithering; = 0.5 all other times.

out: itable The nearest location in the color look-up table.

real: r, g, b, errfac

integer: itable
**DECLARATION:**  Subroutine LCUC (text, nchar)

**FUNCTION:**  Converts a text string to be all upper case.

**PARAMETERS**

`in:`  nchar  Number of characters in text.

`text`  Text string to be converted. The text string will be modified on return.

`integer: nchar`  

`character: text*73`
DECLARATION: Function LENTXT (text, len)

FUNCTION: Determines the length of a text string.

PARAMETERS

  in:  text  Text string.
       len  The dimensioned length of the text string.

  out:  lentxt  Location of the last non-blank character in the text string.

integer: len, lentxt

character: text*73
DECLARATION: Subroutine LITSRC (nlsrce, amount, jlocat, x, y, z)

FUNCTION: Sets light source parameters.

PARAMETERS

in: nlsrce  Light source number.

amount  Light source intensity. If the amount is zero, the other parameters are ignored. Amount is a value between 0 and 1.0.

jlocat  - 0 if light source is at the eye of the observer (x,y,z are ignored).

- 1 if one light source is at infinity (x,y,z are components of a vector pointing from the origin in the light source direction).

- 2 if the light source is at the user-defined position (x,y,z are the coordinates of the light source position).

x  x coordinate of light source.

y  y coordinate of light source.

z  z coordinate of light source.

real: amount, x, y, z

integer: nlsrce, jlocat

COMMON BLOCKS: /CHFWAS/
/CLIMIT/
/CMEMRY/

SEE ALSO: INQLIT
**DECLARATION:** Subroutine LODCOL (ipart, prtcol, colors)

**FUNCTION:** Loads color information for part number being processed.

**PARAMETERS**

in: ipart  
Part number currently being processed. This number should be equivalent to the number that is passed into subroutine PIPLIN. Displacement arrows are loaded with part numbers starting at 10001.

prtcol  
Reference array of rgb colors for each part in the picture. This array should be equivalent to the array that is passed into subroutine HIDLIN.

colors  
r,g,b color components of part ipart.

real: prtcol(3,:), colors(3)

integer: ipart

**SEE ALSO:** HIDLIN
DECLARATION: Subroutine MAPWV (sx, sy, wxmin, wymin, ivxmin, ivymin, x, y, iy)

FUNCTION: Transforms world coordinates into screen coordinates. X and y coordinates should be clipped so they are visible on the screen.

PARAMETERS

in: sx Window to viewport scale factor in x.
    sy Window to viewport scale factor in y.
    wxmin Minimum x-coordinate for window in world.
    wymin Minimum y-coordinate for window in world.
    ivxmin Minimum x viewport boundary (0 to 1).
    ivymin Minimum y viewport boundary (0 to 1).
    x x world coordinates that is modified on return to x screen coordinate.
    y y world coordinates.

out: iy y screen coordinate rounded to the nearest integer

real: sx, sy, wxmin, wymin, ivxmin, ivymin, x, y

integer: iy

SEE ALSO: INQSCL INQPRT GETWND
**DECLARATION:**
Entry MAXMIN (npart, limprt, tm, displc, warp)

**FUNCTION:**
Finds and stores the maximum and minimum coordinates of a part.

Routine compares maximums and minimums to previous limits found from other parts processed by MAXMIN.

**PARAMETERS**
in:
- npart: Part number to process.
- limprt: Part limits array.
- tm: Transformation matrix for part npart.
- displc: Displacement factor (if used).
- warp: Function warping value (if used).

real: tm(4,4), displc, warp(3)

integer: npart, limprt(2,maxitm)

**COMMON BLOCKS:**
/CLIMIT/
/MASTER/

**SEE ALSO:** GETLIM
**DECLARATION:** Subroutine MODIFY (cpynam, icode, value)

**FUNCTION:** Modifies transformation matrix or attribute list for copy cpynam.

**PARAMETERS**

**in:**
- **cpynam** Copy name of instance to transform or change attribute list.
- **icode** Modification flag.

- 1 if initialize copy's transformation matrix.
- 2 if reorient copy to local axis system.
- 3 if scale copy.
- 4 if rotate copy.
- 5 if translate copy.
- 6 if clearing attribute list for copy.
- 7 if setting polygon order for copy.
- 8 if setting copy color.
- 9 if setting feature angle.
- 10 if setting shrink factor.
- 11 if setting shading type.
- 12 if setting shadow casting.
- 13 if setting transparency parameters.
- 14 if setting contour flag.
- 15 if setting fringe flag.
- 16 if setting diffused light intensity.
- 17 if setting displacement factor.
- 18 if setting post flag.
- 19 if setting light source parameters.
- 20 if setting warp vector.
- 21 if setting dotted hidden lines.
- 22 if setting node numbers flag.
- 23 if setting element numbering flag.
- 24 if setting local origin.
- 25 if setting poorman flag.
- 26 if displaying coordinate triad.
- 27 if setting displacement vector arrows scale factor.

**value** Values for transformation or attribute modifications.

**real:** value(*)

**integer:** icode

**character:** cpynam*(*)
COMMON BLOCKS: /CHFWAS/
/CLIMIT/
/CMEMRY/
DECLARATION: Subroutine MOV4X4 (a, b)

FUNCTION: Copies the first matrix into the second matrix.

PARAMETERS

in:  
a    First 4x4 matrix.
b    Second 4x4 matrix.

real:  a(4,4), b(4,4)
DECLARATION: Subroutine MOVABS (ix, iy)

FUNCTION: Update the current point to the specified screen coordinates.

PARAMETERS

in: ix, iy Screen coordinates of the desired point. (0.0) is the lower left point of the screen.

integer: ix, iy
DECLARATION: Subroutine MUL4X4 (a, b, c)

FUNCTION: Performs the matrix multiply, \( a \times b = c \).

PARAMETERS

in:  
  a    First 4x4 matrix.
  b    Second 4x4 matrix.

out:  
  c    Resulting 4x4 matrix.

real:  
  a(4,4), b(4,4), c(4,4)

SEE ALSO: INT4X4
          MOV4X4
DECLARATION: Entry NEWGRP (numgru)

FUNCTION: Sets up a new active group.

PARAMETERS

in: numgru  Group number of new active group.

integer: numgru

SEE ALSO: GETAGR
DECLARATION: Subroutine NORMAL (nedges, coords, xcentr, xn, contst)

FUNCTION: Calculates normals at nodes of polygon.

WARNING: The coordinate array should close the polygon.

PARAMETERS

in:  
nedges  Number of vertices in polygon.
coords  Coordinate array for polygon.
xcentr  Coordinate array of center point.

out:  
   xn  Normal array for polygon.
contst  .TRUE. if polygon is concave.

real:  
   xn(3,*), coords(3,*), xcentr(3)

integer:  
nedges

logical:  
   contst
DECLARATION: Subroutine NORMTM (tm, tmnorm)

FUNCTION: Normalize the columns of a transformation matrix.

PARAMETERS

in: tm A transformation matrix.

out: tmnorm A normalized transformation matrix.

real: tm(4,4), tmnorm(4,4)

SEE ALSO: PIPSRC
DECLARATION: Subroutine NORVEC (vector)

FUNCTION: Normalize a vector.

PARAMETERS
  in: vector   The vector to be normalized.
  real: vector(3)
DECLARATION: Subroutine NRMAVE (nedges, polnor, avenor)

FUNCTION: Computes the average normal of all the normals of a polygon.

PARAMETERS

in:  nedges  Number of vertices of the polygon.
     polnor  Unaveraged normal array for polygon nodes (assumed to be unit normals).

out: avenor  Average unit normal of the polygon.

real: polnor(3,nedges), avenor(3)

integer: nedges
DECLARATION: Subroutine OPNFIL (prmpt, pstrng, fileid, stat, acc, frm, irec, iunit, ierror)

FUNCTION: Opens a disk file.

PARAMETERS

in: prmpt .TRUE. if filename should be prompted for.
     .FALSE. if fileid should be used as the filename.
     pstrng Character string containing prompt.
     fileid Character file id if prompt = .FALSE.
     stat Status of the file to be opened. Must be an allowable file status, such as 'old', 'new', 'scratch' or any other allowable status.
     acc Access of the file to be opened. Must be 'sequential', or 'direct'.
     frm Format of the file to be opened. Must be an allowable file format, such as 'formatted', 'unformatted', or 'binary'.
     irec Record length for direct access files.

out: iunit Device logical number of file opened.
      ierror Error flag.
      -1 if OPNFIL was run successfully.
      -0 if OPNFIL was not run successfully.

integer: irec, iunit, ierror

logical: prmpt

character: fileid(*), pstrng(*), stat(*), frm(*), acc(*)
OUTBOX

DECLARATION: Subroutine OUTBOX (npart, tm, atribt)

FUNCTION: Displays a bounding box around part npart.

PARAMETERS

in: npart Part number to display bounding box around.
    tm Transformation matrix.
    atribt Part attribute list information for part npart.

real: tm(4,4), atribt(*)

integer: npart

COMMON BLOCKS: /CHFWAS/
                /CLIMIT/
                /CMEMRY/
                /MASTER/
                /CATTRI/
DECLARATION: Subroutine PCHECK (nedge, xx, nornew, jconn, jpoi, numply)

FUNCTION: Divides a concave polygon into numply convex polygons.

It is possible that warped concave polygons will not result in a division into convex polygons.

PARAMETERS

in:  nedge Number of sides in original polygon.
nornew Normals at the nodes of the original polygon.
xx Coordinate array of concave polygon.

out: jconn Connectivity of convex polygons.
jpoi Array of number of edges in new convex polygons.
numply Number of polygons after subdivision.

real: xx(3,*)

integer: jconn(vertmx+1,*), jpoi(vertmx+1), nedge, numply, nornew
DECLARATION: Subroutine PERDST

FUNCTION: Sets window based on perspective angle and distance.

SHOULD ONLY BE CALLED IF AUTOCENTER IS NOT ON.

Should be called if in perspective mode, after you have modified either the distance or perspective angle.
Subroutine PIPLIN (sx, sy, ivxmin, ivymin, wxmin, wxmax, wymin, wymax, zmin, zmax, oldcor, polnor, oldfun, nodes, npolab, tm, atribt, nedges, ipart)

Transforms, clips, and maps polygons to screen coordinates, and outputs lines and polygons in line drawing mode. If hidden line is on, PIPLIN will store polygons and lines for further processing.

**PARAMETERS**

_in:_
- **sx:** Window to viewport scale factor in x direction.
- **sy:** Window to viewport scale factor in y direction.
- **wxmin:** Minimum x window coordinate.
- **wxmax:** Maximum x window coordinate.
- **wymin:** Minimum y window coordinate.
- **wymax:** Maximum y window coordinate.
- **zmin:** Distance from eye to near clipping plane.
- **zmax:** Distance from eye to far clipping plane.
- **polcor:** The coordinates of the polygon/line.
- **tm:** Transformation matrix (This transformation matrix will also transform coordinates from world to eye system).
- **atribt:** Part attribute information for line or polygon.
- **nedges:** Number of vertices in current polygon or line.
- **nodes:** Array of node numbers for the polygon used for labeling.
- **npolab:** Polygon label number.
- **ipart:** Part number that line or polygon came from. Used for hidden line removal.
- **ivxmin:** Left viewport screen coordinate x.
- **ivymin:** Bottom viewport screen coordinate y.
- **oldcor:** The coordinates of the line or polygon.
- **polnor:** Normals of the polygon. Used with the feature option during hidden line.
- **oldfun:** Scalar function values at the vertices. Used for contours.
real: sx, sy, wxmin, wymin, wxmax, wymax, zmin, zmax,
polcor(3,vertmx+1), tm(4,4), atribt(*), ivxmin, ivymin,
oldcor(3,vertmx+1), polnor(3,vertmx+1), oldfun(vertmx+1)

integer: nedges, nodes, npolab, ipart

COMMON BLOCKS: /DEVI/
/CATTRI/
/CFLAGS/

SEE ALSO: PIPSRC
INQSCL
INQPRT
GETWND
INQCNT
GETCLP
NORMAL
DECLARATION: Subroutine PIPSRC (sx, sy, ivxmin, ivymin, wxmin, wxmin, 
wymin, wymax, zmin, zmax, numvrt, polcor, 
polnor, polfun, tm, tmnorm, atribt, 
shdlen, shdmin, shdmax)

FUNCTION: Transforms, clips and maps to screen coordinates, and stores 
polygons for later processing by the hidden surface 
processor.

PARAMETERS

in: sx Window to viewport scale factor in x direction.
sy Window to viewport scale factor in y direction.
wxmin Minimum x window coordinate.
wxmax Maximum x window coordinate.
wymin Minimum y window coordinate.
wymax Maximum y window coordinate.
zmin Distance from eye to near z clipping plane.
zmax Distance from eye to far z clipping plane.
umvrt Number of vertices in this polygon.
polcor Coordinates of the polygon.
tm Transformation matrix. (This transformation matrix 
will also transform coordinates from world to eye 
system.)

atribt Attribute information for the polygon.
shdlen Shadow length.
tmnorm Normalized transformation array.
ivxmin Left viewport screen coordinate x.
ivymin Bottom viewport screen coordinate y.
polnor Normals of the polygon.
warped .TRUE. if checking for warped polygons on edge.
polfun Polygon function values.

real: tm(4,4), atribt(*), polcor(3,*), polnor(3,*), sx, sy, wxmin, 
wymin, wxmax, zmin, zmax, numvrt(4,4), ivxmin, ivymin, 
polfun(*), shdlen

integer: numvrt

logical: warped
COMMON BLOCKS: /DEVI/
/CATTRI/
/CFLAGS/

SEE ALSO: PIPLIN
INQSCCL
INQPRTR
GETWND
INQFRN
GETCLP
NORMAL
PCHECK
NORMTM
POLFIL

DECLARATION: Subroutine POLFIL (nedges, ix, iy, r, g, b)

FUNCTION: Draws a filled polygon.

PARAMETERS

in: nedges Number of edges in polygon to be output.
    ix, iy Screen coordinate arrays of the polygon.
    r, g, b Color components (0.0 - 1.0) of the polygon.

real: r, g, b

integer: nedges, ix, iy
**DECLARATION:**
Subroutine POLSTA (nedges, coords, wxmin, wxmax, wymin, wymax, polxmx, polxmn, polymx, polymn, istat)

**FUNCTION:**
Gets polygon maximums, minimums, and status (for x and y).

All coordinates are in the eye coordinate system.

**PARAMETERS**

**in:**
- **nedges**: Number of vertices in the polygon or line.
- **coords**: Coordinate array for vertices.
- **wxmin**: Minimum x window coordinate.
- **wxmax**: Maximum x window coordinate.
- **wymin**: Minimum y window coordinate.
- **wymax**: Maximum y window coordinate.

**out:**
- **istat**: Status of the polygon.
  - 1 if trivially accept.
  - 2 if trivially reject.
  - 3 if must clip the polygon.
- **polxmx**: Maximum polygon x coordinate.
- **polxmn**: Minimum polygon x coordinate.
- **polymx**: Maximum polygon y coordinate.
- **polynm**: Minimum polygon y coordinate.

**real:**
- **wxmin, wxmax, wymin, wymax, polxmx, polxmn, polymx, polymn, coords (3, nedges)**

**integer:**
- **nedges, istat**

**SEE ALSO:**
- CLPPZE
- CLPPOL
- CLPLIN
- ZPLSTA
DECLARATION: Subroutine POLYS (npart, limits, tm, atribt)

FUNCTION: Takes all polygons in part npart, does pre-processing (such as smooth shading and vector arrows) and dumps them into the appropriate pipeline.

PARAMETERS

in: npart Part number to process.
    limits Part limits array.
    tm Transformation matrix.
    atribt Part attribute list for part npart.

real: atribt(*), tm(*,*)

integer: npart, limits

COMMON BLOCKS: /CATTRI/
                /CLIMIT/
                /CHFWAS/
                /MASTER/
                /MEMRY/

SEE ALSO: PIPLIN
          PIPSRC
**DECLARATION:** Subroutine PORMAN (nedges, coords, order, plot)

**FUNCTION:** Determines if polygons are front or back facing, then flags back facing polygons for elimination.

**PARAMETERS**

*in:*  
nedges Number of vertices for this polygon. 
coords Coordinate information for this polygon. 
order Polygon ordering flag.  
  - 0. if poorman is not on for the polygon being processed.  
  - 1. if clockwise element ordering.  
  - 2. if counterclockwise element ordering. 

*out:*  
plot If .TRUE., the polygon is front facing and therefore visible. 

**real:** coords(3,nedges), order 

**integer:** nedges 

**logical:** plot
DECLARATION: Subroutine PROMPT (txt)

FUNCTION: Outputs a character string to the display device.

PARAMETERS

in: txt Character string that contains the required prompt.
character: txt*80
DECLARATION: Subroutine PRSPC (coord)

FUNCTION: Transforms point to perspective viewing.

PARAMETERS

in: coord  x,y,z coordinate of point to transform according to perspective. The coordinates of the point are modified on return.

real: coord(3)

COMMON BLOCKS: /VEWSTF/

SEE ALSO: PRSTSTF
**DECLARATION:** Subroutine PRSSTF

**FUNCTION:** Sets up perspective parameters according to the look-at and look-from point.

*PRSSTF be called after the look-at and look-from point have been established, and before any calls to PRSPC, PIPLIN, or PIPSRC.*

**COMMON BLOCKS:** /VEWSTF/

**SEE ALSO:** PRSPC
PIPLIN
PIPSRC
**DECLARATION:** Subroutine PURGE (cpynam)

**FUNCTION:** Removes a call to an instance wherever listed in the hierarchical data structure.

**PARAMETERS**

- **in:** cpynam Instance name to be deleted.
- **character:** cpynam(*)
DECLAREATION: Subroutine PVEC (point, ctm, dis, sx, sy, ivxmin, ivymin, wxmin, wymin, wxmax, wymax, zmin, zmax, zrange, polfun, atribt, node, npart, ipc)

FUNCTION: Creates and outputs displacement arrows.

PARAMETERS

in: point Coordinates at base of arrow.
dis Displacement array at point.
sx Window to viewport scale factor in x.
sy Window to viewport scale factor in y.
ivxmin Left viewport screen, x coordinate.
ivymin Bottom viewport screen, y coordinate.
wxmin Minimum x window coordinate.
wxmax Maximum x window coordinate.
wymin Minimum y window coordinate.
wymax Maximum y window coordinate.
zmin Distance from eye to near z clipping plane.
zmax Distance from eye to far z clipping plane.
zrange Distance between z clipping planes.
polfun Polygon function values.
atribt Attribute list of polygon containing point.
node Vertex number of point in polygon.
npart Part number of polygon containing point.
ipc Arrow part number.
ctm Transformation matrix for polygon containing point.

real: point(3), ctm(4,4), dis(3), sx, sy, ivxmin, ivymin, wxmin, wymin, wxmax, wymax, zmin, zmax, zrange, polfun(*), atribt(*)

integer: ipc, node, npart

COMMON BLOCKS: /CATTRI/
/CFWAS/
/CMEMRY/
RDINIT

DECLARATION: Subroutine RDINIT

FUNCTION: Initializes the graphics device, and initializes the common block /RESOLT/.

COMMON BLOCKS: /RESOLT/
DECLARATION: Subroutine READMV (geofil, funfil, dspfil)

FUNCTION: Reads MOVIE.BYU files for database definition.

READMV may be called many times.

PARAMETERS

in: geofil  Geometry file name.
    funfil  Function file name.
    dspfil  Displacement file name.

character: geofil(*), funfil(*), dspfil(*)

COMMON BLOCKS: /CHFWAS/
               /CLIMIT/
               /CMEMRY/
DECLARATION: Entry REDCHK (filnam)

FUNCTION: Reads a checkpoint file.

All database descriptions defined by you before a REDCHK call will be deleted, because the checkpoint file will overwrite the existing database.

PARAMETERS

in: filnam The name of the file to be read in for database definition.

character: filnam(*)

SEE ALSO: WRTCHK
DECLARATION: Subroutine RESET

FUNCTION: Initializes the global transformation matrix.

SEE ALSO: INQGLB
SETGLB
ROTATE
SCALE
**DECLARATION:** Subroutine ROTATE (rx, ry, rz)

**FUNCTION:** Performs global rotate about a right handed coordinate system.

**PARAMETERS**

in:
- **rx** Rotation about the x-axis in degrees.
- **ry** Rotation about the y-axis in degrees.
- **rz** Rotation about the z-axis in degrees.

real: **rx, ry, rz**

**SEE ALSO:** RESET
SETGLB
SCALE
INQGLB

---

Chapter Two  TECHNICAL REFERENCES  2.133
DECLARATION: Subroutine RPLCOP (sordex, ctm, atlist)

FUNCTION: Replaces transformation and attribute information for an instance.

PARAMETERS

in:  
sordex The instance sort number.
ctm Transformation matrix for the instance.
atlist Attribute list for the instance.

real: atlist(maxatt), ctm(4,4)

integer: sordex

COMMON BLOCKS: /CHFWAS/  
/CLLIMIT/  
/CMEMRY/

SEE ALSO: EXTCOP
DECLARATION: Subroutine RPLGLB (glb4x4, glbatt)

FUNCTION: Sets global attributes to those in the list coming in.

PARAMETERS

in:  glb4x4 Global transformation matrix.
     glbatt Global attribute array, defined as follows:

     (1) – Red background intensity (0-1).
     (2) – Green background intensity (0-1).
     (3) – Blue background intensity (0-1).
     (4) – Window x minimum (world coordinate).
     (5) – Window x maximum (world coordinate).
     (6) – Window y center (world coordinate).
     (7) – Perspective angle (degrees).
     (8) – Distance.
     (9) – X look-from point.
     (10) – Y look-from point.
     (11) – Z look-from point.
     (12) – X look-at point.
     (13) – Y look-at point.
     (14) – Z look-at point.
     (15) – Near z clipping plane.
     (16) – Far z clipping plane.
     (17) – X minimum viewport (0-1).
     (18) – X maximum viewport (0-1).
     (19) – Y center viewport (0-1).
     (20) – Near fog plane.
     (21) – Far fog plane.
     (22) – Light number.
     (23) – Light location flag.
     (24) – Light intensity.
     (25) – X position for light.
     (26) – Y position for light.
     (27) – Z position for light.

22 - 27 are repeated for each light source.

real:  glb4x4(4,4), glbatt(*)

COMMON BLOCKS:  /CHFWAS/
                 /CMEMRY/
DECLARATION: Subroutine RUNANI (begfrm, endfrm, ierr)

FUNCTION: Runs the animation.

PARAMETERS

in:  
begfrm  Beginning point to start animation (keyframe.subframe).
endfrm  Ending point to end animation (keyframe.subframe).

out:  ierr  Error flag.
- 0 if the animation was run successfully.
- 1 if the animation was not run successfully.

character:  begfrm*10, endfrm*10

COMMON BLOCKS:  /CMEMRY/
/DEVIL/
/CLIMIT/
/CHFWAS/
DECLARATION: Entry SAVTXT (mesage, xsize, ysize, twodim, xpos, ypos, zpos, xang, yang, zang, r, g, b, xslant, n, nsav)

FUNCTION: Saves text string attributes.

PARAMETERS

in:  
n- Text string number.
mesage- The text string array.
xsize- X dimension factor.
ysize- Y dimension factor.
twodim- If .TRUE., message is 2-D. .False., 3-D text strings.
xpos- X position of text string in world coordinate.
ypos- Y position of text string in world coordinate.
zpos- Z position of text string in world coordinate.
xang- X directional vector.
yang- Y directional vector.
zang- Z directional vector or angle of text string if in 2-d mode.
r, g, b- Red, green and blue color values.
xslant- Rotation about x-axis for 3-d text string in degrees.

out:  
nsav- Total number of stored messages.

real: xsize, ysize, twodim, xpos, ypos, xang, yang, zang, r, g, b, xslant

integer: n, nsav

character: mesage(80)

SEE ALSO: INQTXT
DECLARATION:  Subroutine SCALE (sx, sy, sz)

FUNCTION:  Performs global scale.

PARAMETERS

in:  sx  Scaling in the x direction.
     sy  Scaling in the y direction.
     sz  Scaling in the z direction.

real:  sx, sy, sz

SEE ALSO:  RESET
          ROTATE
          SETGLB
          INQGLB
**DECLARATION:** Entry SETBGC (red, green, blue)

**FUNCTION:** Set background color.

All values are from 0 to 1. Values are adjusted to those actually found in the color look-up table (if one is being used).

**PARAMETERS**

in: red Red intensity of background color.
green Green intensity of background color.
blue Blue intensity of background color.

**SEE ALSO:** SETLUT ITABLE INQBGC
DECLARATION: Entry SETBOX (npart, cormax, cormin)

FUNCTION: Stores the x, y, z limits of a part.

PARAMETERS

  in: npart     Part number.
       cormax    Maximum x, y, and z coordinates of box.
       cormin    Minimum x, y, and z coordinates of box.

  real: cormax(3), cormin(3)

  integer: npart

SEE ALSO: INQBOX
DECLARATION: Entry SETCLP (znear, zfar)

FUNCTION: Sets z clip information.

PARAMETERS

in: znear  Distance from eye to nearest z clipping plane.
      zfar  Distance from eye to farthest z clipping plane.

real: znear, zfar

SEE ALSO: GETCLP
DECLARATION: Entry SETCNT (iflag)

FUNCTION: Sets number of contour levels.

PARAMETERS

in: iflag Number of contour levels to be displayed.
integer: iflag

SEE ALSO: INQCNT
Subroutine SETCOL (r, g, b, errfac)

Updates the current color to the specified color.

in:  r, g, b  New color components (0.0 - 1.0).
     errfac  Equal to 0.0 for dithering; - 0.5 all other times.

real:  r, g, b, errfac
DECLARATION: Entry SETCON (key, sordex, thecnt)

FUNCTION: Sets instance continuity at a keyframe for animation.

PARAMETERS

in: key Key frame number.
sordex Sort index number of the instance.
thecnt The continuity value at keyframe key.

real: thecnt

integer: key, sordex

SEE ALSO: INQCON
DECLARATION: Entry SETDEF (numcol, red, grn, blu)

FUNCTION: Sets the red, green, and blue color components of color number numcol.

All color values are from 0 to 1.

PARAMETERS

in: numcol Color number (1 to 6).
    red Red intensity of default color number numcol.
    grn Green intensity of a default color number numcol.
    blu Blue intensity of a default color number numcol.

real: red, grn, blu

integer: numcol

SEE ALSO: GETDEF
ENTRY: Entry SETDSP (xvec, yvec, zvec)

FUNCTION: Sets the displacement direction cosine components that will be used with displacement fringes or contours.

PARAMETERS

- **in:**
  - xvec: X component of direction vector.
  - yvec: Y component of direction vector.
  - zvec: Z component of direction vector.

- **real:** xvec, yvec, zvec

SEE ALSO: INQDSP
DECLARATION: Entry SETDST (dist)

FUNCTION: Sets the distance from the eye to the look at point.

SETDST is only valid when software is in perspective angle mode, and when auto center is off. The function of this routine can also be obtained by adjusting the look-from point.

PARAMETERS

in: dist Distance from eye to look-at point.
real: dist

COMMON BLOCKS: /VEWSTF/

SEE ALSO: INQDST ATFROM INQFLG
DECLARATION: Entry SETFLG (cflag, lflag)

FUNCTION: Sets global flags.

PARAMETERS

in: cflag  Key to what is being set, which is one of the following:

'ALIA' Anti-aliasing flag.
'SHDO' Shadow Flag.
'AXLS' Axis are to be drawn.
'BOUN' Draw bounding box instead of model.
'BORD' Draw border.
'CENT' Auto center flag.
'COLO' Draw colored lines.
'DISP' Displacements are performed.
'DITH' Dithering status.
'ELEM' Element labeling status.
'FOUR' Four view status.
'GLAS' Transparency status.
'HAZE' Haze/fog option.
'HIDD' Hidden line status.
'LABE' Label contours flag.
'LEGE' Display contour legend.
'NODE' Node labeling status.
'ONED' Warped polygons on edge status.
'OVER' Overlay status.
'PERA' Perspective or windowing toggle status.
'PERS' Perspective/orkographic status.
'POLY' Polygon clipping performed.
'POOR' Poorman status.
'SHAD' Shaded images status.
'SYMM' Symmetric contours or fringes status.
'VECT' Vector arrows status.
'WARP' Function warping status.
'TEXT' Text display status.
'TK41' Tektronix 4129 hardware status.
'LABV' Label for views status.
'PLYF' Polygon fill shaded images status.

iflag  .TRUE. or .FALSE. status of cflag.

out: iflag  .TRUE. or .FALSE. status of cflag.

logical: iflag

character: cflag(*)

COMMON BLOCKS: /CFLAGS/

SEE ALSO: INQFLG
DECLARATION: Entry SETFNC (nlevel, valmin, valmax)

FUNCTION: Sets the number of color fringes to be used, and the minimum and maximum range for fringes.

PARAMETERS

in: nlevel Number of function levels to be used.
    valmin Minimum value of color fringe.
    valmax Maximum value of color fringe.

real: valmin, valmax

integer: nlevel

SEE ALSO: INQFNC
DECLARATION: Entry SETFOG (dist1, dist2)

FUNCTION: Sets the distance to the nearest and farthest fog planes.

PARAMETERS

in: dist1 Distance from eye to nearest fog plane.
    dist2 Distance from eye to farthest fog plane.

real: dist1, dist2

SEE ALSO: INQFOG
DECLARATION: Entry SETFRN (iflag)

FUNCTION: Sets number of fringe levels to be displayed.

PARAMETERS

in: iflag Number of fringe levels to be displayed.

integer: iflag

SEE ALSO: INQFRN
DELRATION: Entry SETGLB (tm)

FUNCTION: Saves the transformation matrix that is to be used as the global matrix.

PARAMETERS

in: tm Transformation matrix.
real: tm(4,4)

SEE ALSO: INQGLB
ROTATE
SCALE
RESET
DECLARATION: Subroutine SETLIN (cstyle)

FUNCTION: Updates the current linestyle to solid or dashed.

PARAMETERS

in: cstyle Desired linestyle:
    = 'd' for dashed.
    = 's' for solid.

character: cstyle(*)
DECLARATION: Subroutine SETLUT (table).

FUNCTION: Loads the color look-up table.

PARAMETERS

in: table The desired look-up table. The default is 'c', which is for a full color-ramped look-up table.

character: table(*)
**DECLARATION:** Subroutine SETMOD (mode)

**FUNCTION:** Sets the display device to either graphics or alpha mode.

**PARAMETERS**

- in: mode   The specified mode:
  - 'graphics' to put device in graphics mode.
  - 'alpha' to put device in alpha mode.

- character: mode(*)
DECLARATION: Entry SETPER (setang)

FUNCTION: Sets the perspective angle to the value specified.

SETPER can only be called if the software is in the perspective angle mode.

PARAMETERS

in: setang Perspective angle to be set in degrees.
real: setang

SEE ALSO: INQPER
INQFLG

Chapter Two TECHNICAL REFERENCES 2.156
DECLARATION: Entry SETPRT (xmin, xmax, ycen)

FUNCTION: Sets viewport boundaries.

SETPRT calculates minimum and maximum y values enforcing a one-to-one aspect ratio. All coordinates are normalized from 0 to 1. SETPRT checks for illegal input, and will set the viewport to the entire screen if illegal input is detected.

PARAMETERS

in: xmin Minimum x coordinate for viewport.
    xmax Maximum x coordinate for viewport.
    ycen Center y coordinate for viewport.

real: xmin, xmax, ycen

COMMON BLOCKS: /RESOLT/

SEE ALSO: INQPR

SAMPLE CALL: Call SETPRT (0,-5,-75)
Call sets up a viewport in the upper quarter of the screen.
DECLARATION: Entry SETRAT (key, sordex, therat)

FUNCTION: Sets instance rate at a keyframe for animation.

PARAMETERS

in: key Frame number.
sordex Sort index number of instance.
therat The rate value at keyframe key.

real: therat

integer: key, sordex

SEE ALSO: INQRAT
DECLARATION: Entry SETREF (ncolor, colfnc)

FUNCTION: Sets function value reference colors.

PARAMETERS

   in: ncolor       Number of reference colors to be used.
       colfnc       Array of colors at each function level.

   real: colfnc (3,maxlev)

   integer: ncolor

SEE ALSO: INQREF
SETSCl

**DECLARATION:** Entry SETSCl

**FUNCTION:** Sets window to viewport scale factors.

**COMMON BLOCKS:** /RESOLT/

**SEE ALSO:** SETPRT
            INQPRt
            GETWND
            SETWND
            INQSCL
DECLARATION: Subroutine SETSCN (iy, ixbeg, ixend, r, g, b, errfac)

FUNCTION: Sends a segment of a scanline to the graphics screen.

The graphics device must be in graphics mode before this routine can be called.

PARAMETERS

in: iy Y value of scanline.
ixbeg Beginning x value of segment.
ixend Ending x value of segment.
r,g,b Color arrays for scanline segment.
errfac Equal to 0.0 for dithering; - 0.5 all other times

real: r(0:ixres), g(0:ixres), b(0:ixres), errfac

integer: iy, ixbeg, ixend

COMMON BLOCKS: /RESOLT/
DECLARATION: Entry SETSPL (key, sordex, thespl)

FUNCTION: Sets instance on spline flags for animation.

PARAMETERS

in: key Key frame number.
sordex Sort index number of the instance.
thespl The spline flag:
- 0 for no spline.
- 1 for spline.

real: thespl

integer: key, sordex

SEE ALSO: INQSPL
**DECLARATION:**  Entry SETSTD

**FUNCTION:**  Sets standard function reference colors.
DECLARATION: Entry SETSUB (keyfrm, number)

FUNCTION: Sets number of subframes between keyframes.

PARAMETERS

in: keyfrm Beginning keyframe number.
    number Number of subframes.

integer: keyfrm, number

SEE ALSO: INQSUB
SETTEN

**DECLARATION:** Entry SETTEN (key, sordex, theten)

**FUNCTION:** Sets instance tension at a keyframe for animation.

**PARAMETERS**

in: key  Key frame number.
    sordex  Sort index number of the instance.
    theten  Tension value at keyframe key.

real: theten

integer: key, sordex

**SEE ALSO:** INQ TEN
DECLARATION: Entry SETWND (xmin, xmax, ycen)

FUNCTION: Sets window bounds.

SETWND calculates minimum and maximum y values enforcing a one-to-one aspect ratio.

PARAMETERS

in: xmin  Minimum x world coordinate for window.
    xmax  Maximum x world coordinate for window.
    ycen  Center y world coordinate for window.

real: xmin, xmax, ycen

COMMON BLOCKS: /RESOLT/

SEE ALSO: GETWND
DECLARATION: Subroutine SHRINK (shrk, nedges, polcor, xcentr)

FUNCTION: Moves the nodes of a polygon toward the polygon center.

The coordinate array should close the polygon.

PARAMETERS

in:  
shrk    Shrink factor:
        0 = no shrink.
        1 = full shrink.

nedges  Number of polygon vertices.

polcor  Coordinate array for polygon.

xcentr  Coordinate array of polygon's center point.

real:  
shrk, polcor(3,nedges), xcentr(3)

integer: nedges
**DECLARATION:** Entry SMOCLR (nodbeg, nodend)

**FUNCTION:** Clears the smooth shading routine.

**SEE ALSO:** SMOSTR

**PARAMETERS**

*in:* nodbeg  Node to begin smooth shading.
    nodend  Node to end smooth shading.

*integer:* nodbeg, nodend

**SEE ALSO:** SMOGET
DECLAARATION: Entry SMOGET (nedges, nodes, polnor)

FUNCTION: Retrieves the smooth normals for a polygon.

Use SMOGET after all of the polygons in a part have been processed with SMOSTR.

PARAMETERS

in:  
  nedges Number of edges in the polygon.  
  nodes Array of node numbers for the polygon.

out:  
  polnor Smooth normal of each polygon vertex.

real:  
  polnor(3,nedges)

integer:  
  nodes(nedges), nedges

SEE ALSO: SMOSTR  
          SMOCLR
SMOSTR

**DECLARATION:** Entry SMOSTR (nedges, nodes, polnor)

**FUNCTION:** Averages normals for smooth shading.

Call SMOCLR before making any calls to SMOSTR. You should call SMOSTR for each polygon in a part.

**PARAMETERS**

- **in:**
  - nedges: Number of edges in the polygon being smoothed.
  - nodes: Array of node numbers for the polygon.
  - polnor: Normal of each polygon vertex.

- **real:** polnor(3,nedges)

- **integer:** nodes(nedges), nedges

**SEE ALSO:** SMOCLR

SMOGET
Entry STCONG (key, icode, thecnt)

Sets continuity value for spline animation of look-at or look-from point.

**PARAMETERS**

- **in:**
  - key: Keyframe number.
  - icode: 1 - look at point, 2 - look from point.
  - thecnt: The continuity value at keyframe key (-1 to 1).

- **real:** thecnt

- **integer:** key, icode
Subroutine STOREL (ipart, nedges, polcor, polnor, polfun, nodes, npolab, atribt)

Loads a polygon for hidden line removal.

INTHLR must be called previous to calling STOREL. HIDLIN should be called after all polygons have been processed with STOREL.

Parameters:

- `ipart`: Part number this polygon comes from.
- `nedges`: Number of edges in the polygon.
- `numpol`: Polygon number currently being processed.
- `polcor`: Screen coordinates of the polygon.
- `polnor`: Polygon normals at the vertices.
- `polfun`: Polygon scalar values at the vertices.
- `nodes`: Node number array for the vertices.
- `npolab`: Polygon label number.
- `atribt`: Part attribute list information for the polygon.

Real:
- `polcor(3,*)`, `polnor(3,*)`, `polfun(*)`, `atribt(*)`

Integer:
- `numpol`, `nedges`, `nodes(*)`

Common Blocks:
- `/CATTRI/`
- `/CFLAGS/`
- `/HIDSTF/`
- `/KEEP/`

See Also:
- INTHLR
- HIDLIN
DECLARATION: Entry STRATG (key, icode, therat)

FUNCTION: Sets rate for global animation at keyframes.

PARAMETERS

in:  key     Keyframe number.
     icode   Key to which global animation parameter is being set:
               1 = Global scales.
               2 = Global rotates.
               3 = Background color.
               4 = Window.
               5 = Perspective angle.
               6 = Distance.
               7 = Look-from point.
               8 = Look-at point.
               9 = Z clipping planes.
              10 = Viewport.
              11 = Fog planes.
              12 = Light #1.
              13 = Light #2.
                     Continued for each light source.
     therat  The rate value at keyframe key (0 to 1).

real:  thespl

integer: key, icode
DECLARATION: Entry STSPLG (key, icode, thespl)

FUNCTION: Sets onspline flag for look-at or look-from point spline animation.

PARAMETERS

in: key    Keyframe number.
      icode  1 - look-at point, 2 - look-from point.
      thespl The spline value at keyframe key (0 or 1).

real: thespl

integer: key, icode
**DECLARATION:** Entry STTENG (key, icode, theten)

**FUNCTION:** Sets tension values for spline animation for look-at or look-from point.

**PARAMETERS**

- **in:**
  - key: Key frame number.
  - icode: 1 = look-at point, 2 = look-from point.
  - theten: The tension value at keyframe key (-1 to 1).

- **real:** theten

- **integer:** key, icode

**SEE ALSO:** INTENG
DECLARATION: Subroutine SUMMRY

FUNCTION: Displays the global attributes set.

COMMON BLOCKS: /CLIMIT/
/DEVI/
/VEWSTF/
/RESOLT/
COMMON BLOCKS: /CATTRI/
/CFLAGS/
/CLIMIT/
/VEWSTF/

(cont.) TEXT
DECLARATION: Subroutine TEXT (mesage, xsize, ysize, twodim, xpos, ypos, zpos, xang, yang, zang, r, g, b, atribt, xslant, glbflg, ierr)

FUNCTION: Generates and displays alphanumeric text.

PARAMETERS

in: mesage The text string to be displayed.
    xsize X dimension factor.
    ysize Y dimension factor.
    twodim If .TRUE., 2-D text strings are generated.
           If .FALSE., 3-D text strings are generated.
    xpos X position of text string in model coordinates.
    ypos Y position of text string in model coordinates.
    zpos Z position of text string in model coordinates.
    xang X directional vector coordinate.
    yang Y directional vector coordinate.
    zang Z directional vector coordinate (if 3-d).
    r, g, b Angle of text string (if 2-d).
    atribt Red, green, and blue color values for text string.
    xslant Attribute list.
    glbflg Rotation about x-axis for 3-d text string in degrees.
    ierr If ierr = 1, then a character not defined in the character font has been entered.

real: xsize, ysize, xpos, ypos, zpos, xang, yzng, zang, atribt(*), xslant, r, g, b

integer: ierr

logical: twodim, glbflg

character: mesage*80
DECLARATION: Subroutine TMPTS (tm, coord)

FUNCTION: Transforms an x,y,z point according to a transformation matrix.

PARAMETERS

in: tm Transformation matrix.
    coord Coordinate to be transformed. The coordinate is modified on return.

real: tm(4,4), coord(3)
DECLARATION: Subroutine TMPTSN (tm, coord)

FUNCTION: Transforms a normal vector by a transformation matrix.

PARAMETERS

in: tm Transformation matrix.
    coord Normal to be transformed.

real: tm(4,4), coord(3)
DECLARATION: Subroutine TRAVRS (root, itask, sgorit, grsort, trndex, sgrcnt, sitcnt, gstart, matrix, sgrind, grp.ptr, sortdx, grprec, stkfms, atlist, stkiaa)

FUNCTION: Traverses the groups and items structure for picture processing.

PARAMETERS

in: root Root group to be displayed.
    itask Task flag:
            0 = auto-center.
            1 = process part for display.
            2 = write out geometry file.
    sgorit Subgroup or subitem flag array.
    grsort Group sort array.
    sgrcnt Subgroup counter array.
    sitcnt Subitem counter array.
    gstart Group start array.
    matrix Instance transformation (4x4) matrixes.
    sgrind Subgroups that have been processed array.
    grp.ptr Group pointer at recursive level array.
    sortdx Sort index array.
    grprec Group recursion counter array.
    stkfms Stack of ctm's array.
    stkiaa Attribute stack.
    atlist Attribute list transformation index array.

real: matrix(4,4,maxins), stkfms(4,4,maxrec),
      atlist(maxiaa,maxins), stkiaa(maxiaa,0:maxrec)

integer: sgorit(maxins), grsort(maxins), trndex(maxins),
         sgrcnt(maxgrp), sitcnt(maxgrp), gstart(maxgrp),
         sgrind(maxrec), grp.ptr(maxrec), sortdx(maxrec),
         grprec(maxgrp), itask, root
COMMON BLOCKS:
/GATTRI/
/CHFWAS/
/CHEMRY/
/DEVI/
/VEWSTF/
DECLARATION: Entry TXTDIS

FUNCTION: Displays existing text strings.

SEE ALSO: GENTXT
TXTINT
DECLARATION: Entry TXTINT

FUNCTION: Initializes the text string routines.

SEE ALSO: TXTDIS
          GENTXT
**DECLARATION:** Subroutine WARPOL (nedges, tmpcor, tmpnor, iconn, ipoin, numpol)

**FUNCTION:** Subdivides warped polygons that are partly front and partly back facing to the observer.

**PARAMETERS**

- **in:**
  - nedges: Number of vertices in the polygon.
  - tmpcor: Coordinate array of warped polygon.
  - tmpnor: Normals array of warped polygon.

- **out:**
  - iconn: Connectivity array after subdivision of warped polygon.
  - ipoin: Number of vertices in each subdivided polygon.
  - numpol: Number of polygons that the warped polygon was divided into.

- **real:**
  - tmpcor(3,*), tmpnor(3,*)

- **integer:**
  - nedges, iconn(vertmx+1,*), ipoin(*), numpol
DECLARATION: Subroutine WRITMV (npart, limits, tm)

FUNCTION: Stores geometry information for writing a geometry file.

PARAMETERS

| in:  | npart   | Number of parts to be written to the geometry file. |
|      | limits  | Part limits array.                                  |
|      | tm      | Transformation matrix for npart.                    |

| real: | limits(2,*), tm(4,4) |

| integer: | npart |

COMMON BLOCKS: /MASTER/

SEE ALSO: WRTDRV
           WRTGEO
DECLARATION: Entry WRTCHK (filnam)

FUNCTION: Writes a checkpoint file.

PARAMETERS

in: filnam The name of the file to be used as the checkpoint file.

class: character: filnam(*)

SEE ALSO: REDCHK
DECLARATION: Entry WRTDRV (geonam)

FUNCTION: Opens file and initializes variables for a geometry file write.

PARAMETERS

in: geonam Name of the geometry file to be written.

character: geonam(*)

SEE ALSO: WRTGEO
DECLARATION: Entry WRTGEO

FUNCTION: Writes out the geometry file after a tree traversal.

SEE ALSO: WRTDRV
**DECLARATION:** Subroutine ZPLSTA (nedges, coords, zmin, zmax, polzmx, polzmn, istat)

**FUNCTION:** Gets polygon maximums, minimums, and status for z-clipping.

*All coordinates are in the eye coordinate system.*

**PARAMETERS**

**in:**
- nedges: Number of vertices in the polygon or line.
- coords: Coordinate array for vertices.
- zmin: Distance to nearest clipping plane.
- zmax: Distance to farthest clipping plane.

**out:**
- istat: Status of the polygon:
  - 1 if trivially accept.
  - 2 if trivially reject.
  - 3 if must clip the polygon.
- polzmx: Maximum polygon z coordinate.
- polzmn: Minimum polygon z coordinate.

**real:**
- coords, zmin, zmax, polzmx, polzmn, coords (3,nedges)

**integer:**
- nedges, istat

**SEE ALSO:**
- CLPPZE
- CLPPOL
- CLPLIN
- POLSTA
This chapter describes all of the common blocks used in the Raster Graphics Subroutine Package. They are listed in alphabetical order. For each common block, each variable is defined, and the routines accessing the common block are listed.

Information about common blocks is included so you will be aware of common coupled routines. For some applications, you will need to set up the common blocks before an application can call the subroutines that use them.

Common Blocks for Internal Use ONLY

Three common blocks are strictly for internal use. For these common blocks, we have only given the declaration. These common blocks are:

/HIDSTF/
/PIXSTF/
/ZBUFFER/

Make sure you don’t use these common block names in subroutines you supply yourself.

If you want more information about these common blocks, contact the Engineering Computer Graphic Lab at Brigham Young University. The address is in Appendix C.
COMMON BLOCK: /CATTRI/ (IORDER, IDFFUS, ICOLOR, INNODE, INPOLY,
IFEATR, ISHRNK, IWARP, ICONTR, IDASHL,
ISHADE, ISHADO, IGLASS, IFRNGE, ILIGHT,
IDSPLC, IPOST, IFAST, IDRWAX, IAXORG,
IROTAX, IVECTR)

FUNCTION: Contains pointers into the instance attribute array. /CATTRI/ is set up in the subroutine GRAINT.

PARAMETERS:

Points to the:

integer:
IORDER Polygon ordering information.
IDFFUS Diffused light.
ICOLOR Color information.
INNODE Node numbering flag.
INPOLY Polygon numbering flag.
IFEATR Feature angle.
ISHRNK Shrink factor.
IWARP Warp vector.
ICONTR Contour information.
IDASHL Dashed line flag.
ISHADE Shading type flag.
ISHADO Shadow flag.
IGLASS Transparency information.
IFRNGE Fringe flag.
ILIGHT Light information.
IDSPLC Displacement factor.
IPOST Post flag.
IFAST Poorman flag.
IDRWAX Draw local axis flag.
IAXORG Local axis origin.
IROTAX Local axis rotation angles.
IVECTR Displacement vector arrows Scale factor.
### Routines that contain `/CATTRI/`

| ROUTINE | MODIFICATION
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COLORS</td>
<td>MODTFM</td>
</tr>
<tr>
<td>DEFLT</td>
<td>OUTBOX</td>
</tr>
<tr>
<td>DEFPRT</td>
<td>PIPLIN</td>
</tr>
<tr>
<td>DSPTXT</td>
<td>PIPSRC</td>
</tr>
<tr>
<td>GLBAXS</td>
<td>POLOTL</td>
</tr>
<tr>
<td>GRAINT</td>
<td>POLYS</td>
</tr>
<tr>
<td>LIGHTS</td>
<td>PVEC</td>
</tr>
<tr>
<td>LOCAXS</td>
<td>SHOATR</td>
</tr>
<tr>
<td>MODATT</td>
<td>TRAVIN</td>
</tr>
<tr>
<td>MODFRM</td>
<td>TRAVRS</td>
</tr>
</tbody>
</table>
COMMON BLOCK: /CFLAGS/ (POLYGN, POORMN, HIDDEN, IALIAS, DITHER, LCOLOR, FOG, LCONTR, LFRING, LGLASS, LS, SHADW, PERSPC, WARPED, LABNOD, Labele, FSTHID)

FUNCTION: Contains system global flags.

PARAMETERS

.logical: .TRUE. if doing:

POLYGN Polygonal data.
POORMN Poorman.
HIDDEN Hidden line removal.
IALIAS Anti-aliasing.
DITHER Dithering
LCOLOR Colored lines.
FOG Fog/haze.
LCONTR Contours.
LFRING Fringes.
LGLASS Transparency.
LSHADW Shadows.
PERSPC Perspective.
WARPED .TRUE. if checking for warps on edge polygons.
LABNOD Node labeling.
Labele Element labeling.
FSTHID Polygon fill.

ROUTINES THAT CONTAIN /CFLAGS/

CHKPNT EDGSHD PAINT
COLORS FLAGIT PIPLIN
DEBSOR FOURVW PIPSRC
DISALL GLBAXS POLOTL
DSPTXT INTSEC POLOUT
EDGMAK LOCAKS PREALI
EDGMKP MAKSHA SEGGEN

Chapter Three COMMON BLOCKS 3.4
COMMON BLOCK: /CHFWAS/ (ILSINT, ILSLOC, ILSXYZ, ILSNEW, ISHDIN, IPRCOL, ICOPIN, ITRNDE, ISGORI, IGRPDE, ISGITO, IGRSOR, IGRUPA, ICHIDA, IORGPA, ISORDX, INCOPY, INPL, IALIST, IMATRI, ISGRCN, ISITCN, IGSTAR, ITSGIC, ISGRIN, IGRPPT, ISORTD, IGRPRE, ISTKFM, IAASTK, IGRNAM, ITMNUM, ICPNAM, IPSTFL, IFWANI, IANIGL)

FUNCTION: Contains pointers into memory.

PARAMETERS

Points to the:

integer:

- ILSINT Light source intensities.
- ILSLOC Light source location flags.
- ILSXYZ Light source coordinates.
- ILSNEW Modified light source coordinates.
- ISHDIN Shadow intensities.
- IPRCOL Part color array used in the hidden line removal algorithm.
- ICOPIN Instance copies.
- ITRNDE Transformation index.
- ISGORI Child numbers.
- IGRPDE Parent group.
- ISGITO Child flag array.
- IGRSOR Group sort array.
- IGRUPA Root group array.
- ICHIDA Child array.
- IORGPA Item or group flag array.
- ISORDX Sort array.
- INCOPY Copy number.
- INPL Part limit array.
- IALIST Instance attribute lists.
- IMATRI Instance transformation matrices.
- ISGRCN Subgroup counter array.
- ISITCN Subitem counter array.
- IGSTAR Group start array.
- ITSGIC Child test array.
ISGRIN Subgroup counter array.
IGRPPT Group pointer array.
ISORTD Recursion sort index array.
IGRPRE Group recursion test array.
ISTUFM Transformation stack array.
IAASTU Attribute stack array.
IGRNAM Group names array.
ITMNAM Item names array.
ICPNAM Instance names array.
IPSTFL Instance post flag array.
IFWANI Starting unit number for instance animation.
IANTGL Unit number for global animation.

ROUTINES THAT CONTAIN /CHFWAS/

ADDCOL GETITM INTHDP PURGE
ADDFRM GLBACT INTID PVEC
ANIDRV GRAINT INTSHA READMV
BEGANI HIDSUR LITSRC REDGLB
CLRANI HUDGRP MAKSHA RPLCOP
COLORS HUDITM MODIFY RPLCOR
COPNUM HUGCOP NEWANI RPLGLB
DISALL HUGSOR OPNANI RUNANI
FRMONE HUSORT OUTBOX SHOATR
GETATT HUSTA POLYS TRAVIN
GETCOP INQLIT POSFRM TRAVRS
GETGRU INQUIR PRGRUN WRTGLB
COMMON BLOCK: /CLIMIT/ (MAXITM, MAXINS, MAXGRP, MAXMEM, MAXNOD, MAXPOL, MAXVER, MAXREC, MAXUSE, MAXCAL, MAXCHR, MAXIAA, MAXLIT, NP, NJ, NPT, NCON, NPNEW)

FUNCTION: Contains system limits.

PARAMETERS

integer: MAXITM Maximum number of items allowed.
           MAXINS Maximum number of instances allowed.
           MAXGRP Maximum number of groups allowed.
           MAXMEM Maximum amount of memory allocated.
           MAXNOD Maximum number of nodes allowed.
           MAXPOL Maximum number of polygons allowed.
           MAXVER Maximum number of vertices allowed.
           MAXREC Maximum number of recursion levels allowed.
           MAXUSE Maximum amount of memory in use.
           MAXCAL Maximum number of links in tree structure.
           MAXCHR Maximum number of characters in a name.
           MAXIAA Maximum number of words in an attribute list.
           MAXLIT Maximum number of lights.
           NP Number of parts read in.
           NT Number of nodes read in.
           NPT Number of polygons read in.
           NCON Number of nodes in connectivity list.
           NPNEW New number of parts found following tree traversal.
## ROUTINES THAT CONTAIN /CLIMIT/

<table>
<thead>
<tr>
<th>ACTIVE</th>
<th>DISALL</th>
<th>HUPITM</th>
<th>MAKSHA</th>
<th>RPLCOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDCAL</td>
<td>DSPTXT</td>
<td>HUSOR</td>
<td>MODATT</td>
<td>RPLCOR</td>
</tr>
<tr>
<td>ANILIT</td>
<td>EXTCOP</td>
<td>INILIT</td>
<td>MODIFY</td>
<td>RUNANI</td>
</tr>
<tr>
<td>BEGANI</td>
<td>FRMONE</td>
<td>INITGI</td>
<td>MODFMT</td>
<td>SHDHEX</td>
</tr>
<tr>
<td>CATIAA</td>
<td>GEOMOV</td>
<td>INTCAL</td>
<td>OUTBOX</td>
<td>SHOATR</td>
</tr>
<tr>
<td>CHKPNT</td>
<td>GLBAXS</td>
<td>INTSHA</td>
<td>POLYS</td>
<td>SUMMRY</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>GRAINT</td>
<td>LIGHTS</td>
<td>PRGRUN</td>
<td>TRAVIN</td>
</tr>
<tr>
<td>DEFINE</td>
<td>HUACAL</td>
<td>LIMITS</td>
<td>PURGE</td>
<td>TRAVRS</td>
</tr>
<tr>
<td>DEFPRT</td>
<td>HUDCAL</td>
<td>LITSRC</td>
<td>READMV</td>
<td>TRVCOP</td>
</tr>
<tr>
<td>DELETE</td>
<td>HUPGRP</td>
<td>LOCAKS</td>
<td>REDLIT</td>
<td>WRTLIT</td>
</tr>
</tbody>
</table>
COMMON BLOCK: /CMEMRY/ (MEMORY(MXMEM))

FUNCTION: Contains the data base for the program.

You should use /CHFWAS/ as pointers into memory to extract information.

PARAMETERS

integer/real: MEMORY The data base.

ROUTINES THAT CONTAIN /CMEMRY/

ADDCALL
ADDFRM
ANIDRV
BEGANI
CHKPNT
CIRANI
COLORS
COPNUM
DISALL
EXTCOP
FRMONE
GETATT
GETCOP
GETGRU
GETITM
GLBACT
GRAINT
HIDSUR
HUDGRP
HUDITM
HUGCOPI
HUGSOR
HUSORT
HUSSTA
INQLIT
INQUIR
INTHDP
INTHID
INTSHA
LITSRC
MAKSHA
MODIFY
NEWANI
OPNANI
OUTBOX
POLS
POSFRM
Purge
PVEC
READMV
REDGLB
RPLCOP
RPLCOR
RPLGLB
RUNANI
SHOATR
TRAVIN
TRAVRS
WRTGLB
In the hidden algorithm:

**COMMON BLOCK:** /HIDSTF/ (EDGPNT, IOFFST, IBUCKY, IFREE, SHDPNT, IOFSHD, ISHADO, ISHADY, IAVAIL)

In the hidden line algorithm:

**COMMON /HIDSTF/**

& IPOLST(MAXPOL+1),
& XHOLD(MAXEDG),
& YHOLD(MAXEDG),
& ZHOLD(MAXEDG),
& CCONT(MAXEDG),
& NODNUM(MAXEDG),
& SURF(11,MAXPOL),
& PGRID(3,MAXCOR),
& JNUMBR(MAXCOR),
& JVECLS(2,MAXVEC),
& JVECVP(MAXVEC),
& JVTYPE(MAXVEC),
& VECTOR(9,MAXVEC),
& JVP(4,MAXVEC),
& JXXS(MXGRID,MXGRID,MXBSUR),
& JXIV(MXGRID,MXGRID,MXBVEC),
& XLINE(4,MAXSEG)

**FUNCTION:** For internal use for the hidden line and hidden surface algorithms.

*If you want more information on this common blocks should contact The Engineering Computer Graphics Lab, Brigham Young University.*

**ROUTINES THAT CONTAIN HIDSTF:**

- CLENUP
- DEBSOR
- DMPVEC
- EDGMPK
- GENCNT
- GLSSRT
- HIDLIN
- HIDSRP
- HIDSUR
- HSHECG
- HSHNOD
- INTHD2
- INTHDP
- INTHID
- INTHLR
- LBLNOD
- LODCNT
- LODINA
- LODSHD
- PREALI
- SEGGEN
- UPDATE
- LBLELE
COMMON BLOCK: /KEEP/ (NUMPOL, NUMPLS)

FUNCTION: Contains the internal polygon numbers for opaque and shadow polygons used for hidden surface and line removal.

PARAMETERS

integer: NUMPOL Internal opaque polygon number.
         NUMPLS Internal shadow polygon number.

ROUTINES THAT CONTAIN /KEEP/

HIDLIN
INTHLR
INTHDP
INTHID
POLOUT
SHDOUT
COMMON BLOCK: /MASTER/ (COORDS, ICONEC, NUMVRT, FUNVAL, DSPVAL)

FUNCTION: Contains geometric information.

PARAMETERS

real: COORDS Coordinate Array.
integer: ICONEC Connectivity Array.
NUMVRT Number of vertices in each polygon array.
real: FUNVAL Function values at the nodes.
DSPVAL Displacement values at the nodes.

ROUTINES THAT CONTAIN /MASTER/

CHKPNT
GEOMOV
LIMITS
OUTBOX
POLYS
SUMMARY
WRITMV
COMMON BLOCKS: /PIXSTF/ (IBUCKX, IFREEX)

FUNCTION: For internal use in the hidden surface removal algorithm during anti-aliasing.

*If you want more information on this common block, contact Engineering Computer Graphics Lab, Brigham Young University.*

ROUTINES THAT CONTAIN /PIXSTF/

ALIGLS
DEBSOR
INTHID
STRPIX
COMMON BLOCK: /RESOLT/ (IXRES, IYRES)

FUNCTION: Contains the x and y resolution of the display device.

/RESOLT/ is set up in RDINIT.

PARAMETERS

integer: IXRES The x resolution of the display device. Resolution starts at zero, so a device with 1024 pixels should have IXRES = 1023.

IYRES The y resolution of the display device.

ROUTINES THAT CONTAIN /RESOLT/

CENTER HIDSUR SUMMRY
CHKPNT LODINA VEWPRT
FOURVW LODSHD WINDOW
GLBAXS PAINT WRTGLB
COMMON BLOCK: /VEWSTF/ (AT, FROM, ZO, VIEWMT)

FUNCTION: Contains the viewing parameters.

PARAMETERS

real:  

AT  The look-at point.
FROM The look-from point.
ZO  The distance to the perspective projection plane.
VIEWMT The viewing transformation matrix.

ROUTINES THAT CONTAIN /VEWSTF/

ATFROM INTLIT
CENTER PRSPC
CHKPNT PRSTSF
DSPTXT SHDCTR
GLBAXS SHDHEX
GRAINT TRAVRS
INQLAF
COMMON BLOCK: /ZBUFER/ IBUF

FUNCTION: For internal use in the hidden surface algorithm.

If you want more information on this common block, contact Engineering Computer Graphics Lab, Brigham Young University.

ROUTINES THAT CONTAIN /ZBUFER/

ALIGLS
CLRBUF
DEBSOR
INSSEG
PAINT
SHADOW
Chapter Four

INCLUDE FILES

This chapter explains the content of each of the include files needed for the Raster Graphics Display Library. The include files contain parameters controlling site and global control, which you should edit to fit your needs.

The subroutine package performs error checking on all arrays that are dimensioned by the parameters in the include files. Error messages will refer you to specific parameters in an include file.
INCLUDE BLOCK: ANIM.INC (MAXFRM, ISTART, ISTGLB, MAXCOM)

FUNCTION: Contains the variables that define parameters for animation.

PARAMETERS
MAXFRM The maximum number of key frames that can be defined.
ISTART Starting logical unit number for disk access for instance animation.
ISTGLB Logical unit number for disk access for global animation.
MAXCOM The maximum number of commands defining a key frame.

ROUTINES THAT USE ANIM.INC.
ANIDRV
CHKPNT
CTRLGL
DOSPLN
GETATT
MODGLB
MOVMT
OPNANI
POSFRM
RUNANI
SPLNGL
INCLDUE BLOCK:  CVER.INC (VERTMX)

FUNCTION:  Contains the variable that defines the maximum number of vertices in a polygon.

PARAMETERS

VERTMX  The maximum number of vertices in a polygon.

ROUTINES THAT CONTAIN CVER.INC.

CHECK  GLBAXS  PIPLIN  SHDHEX
CLPLZE  HIDSUR  PIPSRC  SHDOUT
CLPPOL  LINSEC  POLYS  SHDTRN
CLPPZE  LOCAXS  PRESUR  SPLITP
DSPTXT  MAKSHA  PROCHL  STACK
GRAINT  OUTBOX  ROLL  TESTP
GENCMT  PCHECK  SEND  WARPOL
INCLUDE BLOCK: FNCT.INC (MAXLEV, MXCLEV, MXCSEG)

FUNCTION: Contains variables that control fringes and contours.

The software will print error messages if these parameters are exceeded.

PARAMETERS

MAXLEV Maximum number of function levels that can be defined for fringes and contours.

MXCLEV Maximum number of separate contour strings at the same contour level.

MXCSEG Maximum number of contour vectors in a contour level.

ROUTINES THAT CONTAIN FNCT.INC:

AFNCTN DRVTXT LODCNT
CHKPNT FRNBAR ORDCNT
CNTBAR FUNCTN PRGRUN
COLORS GENCNT PROCHL
DMPVEC HIDLIN PVEC
INCLUDE BLOCK: HIDN.INC (IAVVRT, MAXPOL, MAXEDG, MAXCOR, MAXVEC, MXGRID, MXBSUR, MTABLE, NXRESX, MXRESY, MAXFRC, MAXFIL, LITCST, NUMSHD)

FUNCTION: Contains the variables defining maximums for picture processing.

PARAMETERS

IAVVRT Average number of vertices in a polygon (approximate).

MAXPOL Maximum number of polygons in the scene. MAXPOL is used in the visible line and surface algorithms. Regular line drawings are not affected.

MAXEDG Maximum number of edges in the scene. MAXEDG is defined to be MAXPOL*IAVVRT.

MAXCOR Maximum number of vertices (nodes) in the scene.

MAXVEC Maximum number of line segments in the scene. MAXVEC is used for hidden line removal.

MXGRID Maximum number of grid cells in X and Y. MXGRID is used for hidden line removal.

MXBSUR Maximum number of polygons in a bucket. MXBSUR is used for hidden line removal.

MTABLE Maximum length of the hashing table. MTABLE is used for hidden line removal.

MXRESX Maximum resolution of the display device in the x direction.

MXRESY Maximum resolution of the display device in the y direction.

MAXFRC Maximum number of pixel fractions on a scan line for anti-aliasing. MAXFRC is used by the hidden surface algorithm.

MAXFIL Length of the array used for the painters hidden surface algorithm.

LITCST Maximum number of lights casting shadows.

NUMSHD Maximum number of shadow polygons that must be stored in the shadow edge list array.
**ROUTINES THAT CONTAIN HIDN.INC.**

<table>
<thead>
<tr>
<th>ROUTINE</th>
<th>ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANTALI</td>
<td>DMPVEC</td>
</tr>
<tr>
<td>CLENUP</td>
<td>EDGMKP</td>
</tr>
<tr>
<td>CLRBUF</td>
<td>FRNBAR</td>
</tr>
<tr>
<td>DEBSOR</td>
<td></td>
</tr>
</tbody>
</table>
INCLUDE BLOCK:

MOV.L.INC (IOUT, IIN, LUN, LUN1)

FUNCTION:
Contains logical unit numbers for file access.

PARAMETERS
IOUT Logical unit numbers for writing to the terminal.
IIN Logical unit number for reading from the terminal.
LUN Logical unit number for disk file access.
LUN1 Another logical unit number for disk file access.

ROUTINES THAT CONTAIN MOV.L.INC:

GETXT
OPNFIL
PROMPT
INCLUDE BLOCK: MSTR.INC (MXNODE, MAXCON, MXPOLY, LITNUM, ITMNUM, INSNUM, IGRPNM, IRECNM, MAXPRT, MAXATT, MXMEM, CONST)

FUNCTION: Contains the variables that define system wide parameters.

PARAMETERS

- **MXNODE**: Maximum number of coordinates that can be read into the data base.
- **MAXCON**: Maximum number of nodes in the connectivity list that can be read into the data base.
- **MXPOLY**: Maximum number of polygons that can be read into the data base.
- **LITNUM**: Maximum number of light sources that can be defined.
- **ITMNUM**: Maximum number of items that can be defined. This corresponds to the maximum number of parts that can be defined.
- **INSNUM**: Maximum number of instances that can be defined.
- **IGRPNM**: Maximum number of groups that can be defined.
- **IRECNM**: Maximum number of levels in the hierarchical tree structure.
- **MAXPRT**: Maximum number of parts in the scene.
- **MAXATT**: Length of the attribute list for an instance.
- **MXMEM**: Length of the array containing the data base.
- **CONST**: A constant to generate standard ASCII numbers for characters.
<table>
<thead>
<tr>
<th>ROUTINES THAT CONTAIN MSTR, INC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDCAL</td>
</tr>
<tr>
<td>ADDFRM</td>
</tr>
<tr>
<td>ANIDRV</td>
</tr>
<tr>
<td>BEGANI</td>
</tr>
<tr>
<td>BNDBOX</td>
</tr>
<tr>
<td>CKKPNT</td>
</tr>
<tr>
<td>CLRANI</td>
</tr>
<tr>
<td>CNTBAR</td>
</tr>
<tr>
<td>COLORS</td>
</tr>
<tr>
<td>COPNUM</td>
</tr>
<tr>
<td>DISALL</td>
</tr>
<tr>
<td>DMPVEC</td>
</tr>
<tr>
<td>DOSPLN</td>
</tr>
</tbody>
</table>

Chapter Four  INCLUDE FILES  4.9
FUNCTION: Contains the variables defining the length of character strings for user responses and database names.

PARAMETERS

LNSIZE The maximum length of an input character string.
MXCHAR The maximum length of a database name.

ROUTINES THAT CONTAIN USER.INC.

GRAINT
SHOATR
This chapter contains the link map for MOVIE.BYU version 6. MOVIE.BYU is a general purpose computer graphics display system that uses RGDLS software. The purpose of this chapter is to aid users that do not have library utilities on their computers.

The link map shows which routines call other routines, and is in the following form:

```
ROUTINE1
    ROUTINE2
        ROUTINE3
        ROUTINE4
    ROUTINE2*
    ROUTINE3
```

Here, the program flows in the following way:

1. ROUTINE1 calls ROUTINE2.
2. ROUTINE2 calls ROUTINE3.
3. ROUTINE3 does not call anything, and returns to ROUTINE2 when finished.
4. ROUTINE2 calls ROUTINE4.
5. ROUTINE4 does not call anything, and returns to ROUTINE2 when finished.
6. ROUTINE2 when finished returns to ROUTINE1.
7. ROUTINE1 calls ROUTINE2 again. The calls that ROUTINE2 makes have already been defined, so the line is tagged with a *.
8. ROUTINE2 when finished returns to ROUTINE1.
9. ROUTINE1 calls ROUTINE3. ROUTINE3 has been defined previously, but does not call any other routines; therefore, ROUTINE3 is shown without the *.
10. ROUTINE3 when finished returns to ROUTINE1, which when finished terminates the program.

MOVIE.BYU contains several routines that are not part of the subroutine library. Be aware that this link map is not for you to actually use—it is just for looking.
CLRANI
   ANFILS
SETMOD
ERASE
SETMOD
PAUSE
   GETXT *
PAUSE *
SETMOD
ERASE
SETMOD
INQPRT
SETPRT
SETMOD
SETPRT
GETAGR
GETGRU *
GETXT *
ACTGRP
   GETGRU *
FOURVW
INQPRT
INQLAF
INQBGC
SETPRT
ATFROM
   INT4X4
   GTUTRN
   GTUROY
   INT4X4
   MUL4X4
   MOV4X4
   GTUROX
   INT4X4
   MUL4X4
   MOV4X4
   MUL4X4
   MOV4X4
DISALL
   PRSSTF
GETAGR
INQCNT
INQFRN
TRAVRS
   INQCLB
   INT4X4
   MOV4X4
   GLBAXS
   MOV4X4
GETWND
INQSCL
INQPRT
GTUTRN
Appendix A

LINK MAP A.7

NORMTM *
ISHDDS
PIPSRC *
PIPLIN *
LODCOL
CNTRPT
SHRINK
CNTRPT
NORMAL
TMPTS
HRDWAR
PIPLIN *
AROCHK
PVEC *
SETARP
OUTBOX
GETWND
GETCLP
INQSCL
INQPRT
INQBOX
PIPLIN *

MOV4X4
GTUTRN
MUL4X4
GTUTRN
LOCAXS *
MAXMIN
TMPTS

WRTGEO

TMPTS
TRAVRS *
CENTER
INQPER
INQGLB
GETLIM
ATFROM *
MOV4X4
GTUTRN
MUL4X4
PRSSTF
TMPTS
PRSPC
SSHDDS
SETCLP
SETWND
GETWND

SHDCTR
INQGLB
GETLIM
MOV4X4
GTUTRN
MUL4X4
TMPTS
SSHDDS
Appendix A  LINK MAP  A.8
STRPIX
CODPS1
STRPIX
CODNG1
STRPIX
CODPS2
STRPIX
PREALI *
ALIGLS
STRPIX
SETONE
GLASS
INSSEG
GETONE
INSSEG
SETONE
INTSEC
STRPIX
SHADOW
ANTALI
GLASS
INSSEG
PAINT
DRWABS
MOVABS
HAZE
DODITH
SETSCN
SETCOL
UPDATE
CLRBUF
CLRNXT
FRNBAR
INQPRT
INQLEV
DODITH
SETSCN
SETCOL
MOVABS
DRWABS
INQBGCC
DSPTXT
GETWND
INQPRT
INT4X4
GTUROZ *
MOV4X4
INQSCL
GTUTRN
INQGLB
MOV4X4
GTUTRN
MUL4X4
PAUSE *
PIPLIN *
CNTBAR
  INQPRT
GETWND
INQLEV
INQBGC
DSPTXT *
SETCOL
MOVABS
DRWABS
INQCNT
HRDWAR
TXTDIS
  INQTXT
    DSPTXT *
  SETMOD
INQPRT
SETMOD
DSPTXT *
SETPRT
ATFROM *
DISALL *
INQPRT
DISALL *
GETXT *
GETCOM
  LCUC
READMV
  GEOMOV
    OPNFIL *
    BNDBOX
    SETFNC
    SETSTD
    ASTSTD
    ASTFNC
    SETFNC
GETAGR
  CHRNUM
GETITM *
ADDCAL
GETITM *
GETGRU *
GETAGR
HUACAL
  HUGCOP
    HUGCO1
  HUSORT
    HUSOR1
  HUSSTA
    HUSST1
HUACAL
  HUGS01
GETCOP *
INTCAL
    INT4X4
  GETXT *
SETPER
ATFROM *
PERDST
INQLAF
INQPER
SETWND
SETWND
INQDST
INQLAF
SETPRT
SETBGC
SETCLP
SETPOG
REDLIT
COMGLB
COPNUM
COPNMI
FRMONE
REDONE
RPLCOP
  COPNUM *
  HUGSOR *
  RPLCP1
MOV4X4
REDGLB *
COMGLB
MODGLB
GTUSCL
  INT4X4
  MUL4X4
  MOV4X4
GTUOX *
GTUROY *
GTUROZ *
ANICOM
MODFRM
GTUTRN
GTUROZ *
GTUROY *
GTUROX *
GTUSCL *
GTUTRN
MOV4X4
RPLGLB
SETGLB
SETBGC
ATFROM *
SETWND
PERDST *
SETPER
SETCLP
SETPRT
SETPOG
ANILIT
MOV4X4
RPLCOP *
SPLNGL
INQFRM
SPLINE
MODGLB *
COMGLB
CTRLGL
GLBCNM
MODFRC
MODGLB *
RPLGLB *
MODFRC
DOSPLN
INQFRM
SPLINE
MODFRM *
ANICOM
MODFRM *
RPLCOP *
FOURVW *
DISALL *
CAMERA
MOV4X4
SPLNGL *
CTRLGL *
MODFRC
MODGLB *
COMGLB
RPLGLB *
DOSPLN *
MODFRM *
DOSPLN *
MODFRM *
ANICOM
RPLCOP *
FOURVW *
DISALL *
CAMERA
MOV4X4
GETXT *
WRTHCK
OPNFL *
GETAGR
INQGLB
INQTXT
INQPER
GETWND
INQPRT
INQBGC
GETCLP
GETDEF
INQBOX
INQFNC
ANQFNC
ANQREF
INQREF
INQFOG
ANFILI
INQFRM
INQSUB
INQ RAT
INQCON
INQ TEN
INQSPL
INTENG
INCONG
INSPLG
INRATG
IOPNAN
ANIDRV
INQFRM
GETXT *
BEGANI
ANFILI
CLRANI *
CNUMOP
ANFILS
OPNANI
GETCOP *
ANFILI
SETFRM
WRTGLB
INQLGB
INQLAF
INQPER
GETWND
INQPRT
INQBGC
GETCLP
INQFOG
WRTLIT
ANFILS
INQFRM
INQSUB
GETXT *
GETNUM
SETSUB
COPNUM *
GETCOP *
INQ RAT
INQ TEN
INQCON
INQSPL
GETNUM
SETRAT
SET TEN
SETCON
SETSPL
INRATG
STRAIG
INTENG
INSPGL
INCONG
STRATG
STTENG
STCONG
STSPGL
INRATG
STRATG
RUNANI *
SETFRM
ANIDRV *
GETXT *
ACTGRP *
DEFINE *
ADDCAL *
CHKNMAM
GETTXT *
GETCOP *
DELETE
GETCOP *
HUDITM
HUSSOR *
HUDCAL
HUSORT *
HUSSTA *
HUDGRP
HUSSOR *
HUDCAL *
PURGE
GETAGR
GETITM *
GETGRU *
GETCOP *
HUPITM
HUDITM *
HUDITM *
HUPGRP
HUDGRP *
CHKNMAM *
MODIFY
GETCOP *
HUSSOR *
ANFILI
NEWANI
OPNANI *
EXTCOP
GETCOP *
HUSSOR *
EXTCPI
GETCOP *
GETAGR
TRAVIN *
WRTONE
MODTFM

Appendix A  LINK MAP  A.16
INT4X4
GTURN
GTUROZ *
GTUROY *
GTUROX *
GTUSCL *
ADDFRM
INQFRM
MODATT
ADDFRM *
CHKNAM *
GETXT *
GETNUM
MODIFY *
GETATT *
SQUISH
GETCOM *
INQLIT
INQUIR
GETCOP *
GETAGR
TRAVIN *
SHOATR
GETTXT *
INQLIT
GETGRU *
GETITM *
GETCOP *
HUCSOR *
GETDEF
SETDEF
SUMMARY
INQLIT
INQFRM
INQCNT
GETTXT *
GETWND
GETCLP
INQLAF
INQPRT
INQBGC
GETLIM
ANQFNC
SQUISH
GETTXT *
GETNUM
SQUISH
ANQDSD
SQUISH
ASTDSP
GETTXT *
GETCOM *
ASTSTD
ANQREF
ASTREF
INQDEV  
INQFRN  
SETFRN  
GETCOM *  
INQDSD  
SQUISH  
SETDSP  
SETFRN  
INQFNC  
SETFNC  
SETSTD  
INQREF  
SETREF  
SETSTD  
INQFOG  
GETCLP  
GLBACT  
SCALE  
INQGLB  
GTUSCL *  
SETGLB  
ROTATE  
INQGLB  
GTUROX *  
GTUROY *  
GTUROZ *  
SETGLB  
SETBGC  
SETWND  
SETPER  
PERDST *  
SETDST  
INQLAF  
ATFROM *  
PERDST *  
ATFROM *  
SETCLP  
SETPRT  
LITSRC *  
SETFOG  
ANFILI  
INQFRM  
GLBACT *  
INQDEV  
INQCNT  
SETCNT  
GETXT *  
GETCOM *  
INQDSD  
SQUISH  
SETDSP  
INQFNC  
SETFNC  
SETSTD  
INQREF  

Appendix A  LINK MAP  A.18
In Chapter One, Example Problems, the fifth and sixth examples refer to this chapter, "An Introduction to Hierarchal Data Structures." We have included this section because reading through it will help you understand the hierarchal data structures used in RGDL.
AN INTRODUCTION TO HIERARCHICAL DATA STRUCTURES

If you read in a geometry file into RGDL using READMV, the program will print a message similar to the following message:

A COPY OF 1 HAS BEEN ADDED

This line with a different number will be repeated for each part the applications program reads from the given geometry file. The message indicates that the hierarchical data structure is doing some default work for you.

You may choose to ignore the hierarchical data structure, but we strongly recommend you learn how to manipulate and use this powerful tool.

This appendix defines three terms you need to know and explains the hierarchical data structure, but does not give examples. You should see Chapter One, especially Examples Five and Six, for examples of how hierarchical data structures are used within RGDL.

Hierarchical Data Structure Terms

Item: An item is a collection of polygons that all behave in the same way. In other words, the polygons share the same attributes and are transformed together. An item cannot be deleted from the database and cannot carry attributes or transformations.

Instance: (or copy) An instance is a copy of an item or a group and can be deleted. It carries attributes as well as transformations. At display time, the instances show up on the screen.

Group: A group is a collection of instances. A group, like an item, cannot be deleted from the database, and it cannot carry attributes or transformations.

What Are Groups And Items?

Groups and items are part of a powerful hierarchical data structure that allows you to create and manipulate parts that may be related to one another.

A simple example of how parts can be created and manipulated is the creation of a car model. The wheels and the car body can be defined as separate items, while the wheels and car body together are a group.

Half of the car body can be modeled. This half can then be copied and mirrored to create an entire car body. A wheel can be created and replicated four times, then placed at the correct locations.
The body and wheels can then be manipulated together as an entire car. All four wheels can be rotated while the car is moving by simply rotating the instance containing the four wheels.

Upon initialization, RGDL creates a default group called ROOT (Capital letters are different from lower case letters. Thus, a group with the name 'ROOT' will be different than another group with the name 'root'). When a geometry file is read in, a copy of each of the parts are placed under the current active group (default ROOT).

Thus, if a three part geometry file were read in, the structure would look like Figure 1.

Because the numbers 1, 2, and 3 are not very descriptive of the geometry, we can change the names to improve the description. For example, we can rename items 1, 2, and 3 to part1, part2, and part3, and change the instance names to copy1, copy2, and copy3. The structure will then look like Figure 2.

Figure 1.

Appendix B HIERARCHICAL DATA STRUCTURES B.3
Because the numbers 1, 2, and 3 are not very descriptive of the geometry, we can change the names to improve the description. For example, we can rename items 1, 2, and 3 to part1, part2, and part3, and change the instance names to copy1, copy2, and copy3. The structure will then look like Figure 2.

As mentioned earlier, only instances can be deleted from the data structure. For example, we can delete copy1 and add another instance of part3. The resulting structure is shown in Figure 3.
As seen in Figure 3, item part1 has simply been unlinked from the data structure. It may be linked up at a later time under any group, or ignored.

Another group called 'NEW GROUP' can be created. Under this group, copies of other groups or items can be added.

The only limit is that the links can not be recursive. For example, a copy of 'ROOT' cannot be added under the group 'ROOT' because a circular list would be created. RGDL checks for recursive links and tells you about them.

A new structure might look like Figure 4.

The term definitions explain that only instances carry transformations and attributes. Transformations concatenate up the tree; thus a rotate of copy2 and a translate of ROOT COPY 1 would result in a scene that has two rotated copy2's, but only one translated copy2 (the rotate would occur before the translate).

In fact, such a scene would contain seven parts: two each of copy2, copy3, and copy4, and one part representing copy1. The one group of copy2, copy3, and copy4 would be translated.
Attributes such as color, shading type, and highlights are not concatenated. Instead they are inherited from "parent" to "child" for attributes not set for the child.

For example, if the color of copy2 was set to red, and the color of ROOT COPY 1 was set to green, the color of copy2 will remain red. If the color of copy3 was not set, it would be green under ROOT COPY 1 because copy3 inherited this color from its "parent".

However, copy3 under ROOT COPY 2 will not inherit color on the way up the tree, but will remain with its default color. An instance that has no attribute after traversing the tree from bottom to top will be assigned a default attribute.

To better understand hierarchical data structures, study the examples shown in Chapter One, especially Examples Five and Six.
Appendix C

SOFTWARE Installation

The magnetic tape supplied with this document is in a PRIME Magsav format for easy installation on other PRIME systems. The magnetic tape contains the following files:

- HIDDEN.F77: RGDL hidden surface routines.
- ANIMATE.F77: RGDL animation routines.
- GRUPIT.F77: RGDL groups and items outlines.
- JONESD.F77: RGDL hidden line routines.
- MAIN.F77: RGDL common use routines.
- MOVLIB.F77: RGDL input/output library routines.

**RGDL device driver routines**
The C routines are for UNIX installations. Only one driver should be linked with an application.

- AED.F77
- AED.C
- RAS110.F77
- RAS120.F77
- RASOUT.C
- T4010.F77
- T4109.F77
- T4111.F77
- T4115.F77
- T4129.F77

**RGDL include files**

- ANIM.INC
- CVER.INC
- FNCT.INC
- HIDN.INC
- MOVL.INC
- MSTR.INC
- USER.INC

**Example problems**

- RSPEX1.F77
- RSPEX2.F77
- RSPEX3.F77
- RSPEX4.F77
- RSPEX5.F77
- RSPEX6.F77

**Example problem geometry file**

- EXP9.GEO
INSTALLATION

RGDL software is for installation on a PRIME computer. Even though the code is generic FORTRAN, it contains include file statements that are system dependent. The device drivers contain code that dumps buffers; this code is also system dependent.

To install RGDL on different operating systems, the include file statements in the source code must be modified to be compatible with your system's requirements. The code to perform system dependent buffer dumping in the device driver should also be changed.

BUGS, PROBLEMS, and COMMENTS

We have done our best to make RGDL and its documentation accurate and clear. However, you may find things that need to be clarified or changed. If you find bugs or problems in the software or the documentation, please contact us. Any positive comments would also be appreciated.

Software Manager
Engineering Computer Graphics Lab
Civil Engineering Department
368 Clyde Building
Brigham Young University
Provo, UT 84602
(801) 378-2812