Index

#pragma directive, 104
$SEGCALL, 85
$SEGRES, 85
  description, 145
_infoblk table, description, 141
_loaded, and soft externals, 106

A

ABS directive, overview, 35
Absolute binary module, definition, 149
ADA, 1
ADDBSS directive, 90
ALIGN
  directive, overview, 47
  pseudo-op, 47
Allocating Central Memory
  Cray PVP system, 99, 100
  default allocation scheme, 98
  definitions of terms, 96
  overview, 95
  segmented programs, overview, 99
  shared-text allocation scheme, 98
  TEXT, DATA, BSS allocation scheme, 98
Assigning modules to segments, 67
Assignment, block & program duplication
  nonsegmented programs, 75
  overview, 75
  segmented programs, 76
Automatic duplicaiton of movable blocks using FLOAT directive, 80

B

Barrier, definition, 149
bin and lib files, exceptions and differences, 24
BIN directive
  example, 125
  with DUPORDER directive, 32
  with FORCE directive, 30
  with NODEFLIB directive, 28
BIN directive (global)
  example, 26
  overview, 26
BIN directive (segment)
  example, 70
  overview, 69
Bin file, definition, 149
Block
  assignment
    and program duplication, 75
    definition, 75
    nonsegmented, 76
  definition, 149
  movable, handling using FLOAT directive, 80
BLOCKDATA subprograms, 25, 30
Branch segment, definition, 55, 56, 149
BSS sections, and shared-text memory allocation, 98

C

C, 1
CAL
  definition, 149
  version 2, 1
Calling tree, precautions for design, 57
CALLXFER directive, overview, 113
Case conversion
  controlling, 51
  convention, vi
CASE directive, overview, 51
CAUTION message, definition, 137
CDBX, definition, 150
Central Memory, allocation of, 95
CF90, 1
CFT77, 1
CODE attributes, and shared-text memory allocation, 98
Command line
  ld(1), 10
  options, -k, 137
  segldr(1), 4
Command line options
  ld(1), 10
  segldr(1), 4

Command options, 15
Commands, explain(1), 138
COMMENT directive, as segment description directive, 66
COMMENT message, definition, 137
Comments
  example, 37
  overview, 36
  using in SEGLDR, 21
Common block
  allocation, ORDER directive, 97
  assignment
    overview, 76
    segmented block, 79
  definition, 150
  duplication in nonsegmented loads, 76
  names, duplication in segmented loads, 76
  naming, directives for
    COMMONS, 31
    FORCE, 30
    MODULES, 30
  reference map, 135
  use and assignment, 75, 79, 83
    block data routines, 84
    data load restrictions, 83
    duplicate common blocks, 77
    referencing data in common blocks, 84
Common block use and assignment, user-assigned common blocks, 83
COMMONS directive (global)
  and duplicated common blocks, 76
  overview, 31
COMMONS directive (segment)
  in segmented loads, 76
  overview, 68
COMPRESS directive, overview, 52
Control, load map, 35
Controlling entry points and execution, 44–47, 113
Controlling entry points and execution, directives for
  CALLXFER, 113
  EQUIV, 45
  overview, 44, 113
  SET, 46
  START, 113
  UNSAT, 46
  XFER, 44
Controlling error messages, 39
Controlling error messages, directives for
  DUPENTRY, 41
  DUPLOAD, 42
  MLEVEL, 39
  MSGLEVEL, 43
  NODUPMSG, 42
  NOUSXMSG, 43
  overview, 39
  REDEF, 40
  USX, 40
Controlling listings, directives for
  Comments, 36
  ECHO, 36
  MAP, 37
  TITLE, 38
  TRIAL, 35
Controlling loading, 47
Conventions, for loader directives, 21
COPY directive, overview, 73
CPUCHECK directive, overview, 51

D

DATA attributes, and shared-text memory allocation, 98
Data loading, definition, 150
Data loads
  and FORCE directive, 80
  and MODULES directive, 80
  restrictions, 83
    block data routines, 84
    referencing data in common blocks, 84
Debug symbol table, 50
Debugging, symbolic
  CDBX definition, 150
  overview, 50, 114
  SYMBOLS directive, 50
Default, directives file, 17
Default directives files, 19
Default libraries, 25
Default system libraries, 18
DEFDIR directive
  example, 111
  overview, 109
DEFHEAP directive, overview, 117
DEFLIB directive
  example, 29
  overview, 29
DEFSTACK directive, 118
DEX, definition, 150
Differences between segldr and ld, 17
Directive termination, 22

Directives
  ABS, 35
  ADDBSS, 90
  ALIGN, 47
  BIN (global), 26
  BIN (segment), 69
  CALLXFER, 113
  CASE, 51
  comments, 36
  COMMONS (global), 31
  COMMONS (segment), 68
  COMPRESS, 52
  conventions, 21
  COPY, 73
  CPUCHECK, 51
  DEFDIR, 109
  DEFHEAP, 117
  DEFLIB, 29
  DEFSTACK, 118
  DUP, 71
  DUPENTRY, 41
  DUPLOAD, 42
  DUPORDER, 32
  DYNAMIC, 90, 91
  ECHO, 36
  ENDSEG, 66
  EQUIV, 45
  FORCE, 30
  FREEHEAP, 119
  HARDREF, 108
  HEAP, 87
  INCFILE, 114
  INCLUDE, 23
  LBIN, 26
  LIB, 27
  LIBDIR, 111
  LINCLUDE, 23
  LLIB, 28
  LOGFILE, 53
  LOGUSE, 53
  MAP, 37
  MLEVEL, 39
  MODULES (global), 30
  MODULES (segment), 67
  MSGLEVEL, 43
  NODEFLIB, 28
  NODUPMSG, 42
  NOUSXMSG, 43
  OMIT, 34
  ORDER, 97
  ORG, 49
  OUTFORM, 112
  PRESET, 48
  REDEF, 40
  SAVE (global), 72
  SAVE (segment), 70
  SCOMMONS (global), 31
  SCOMMONS (segment), 68
  SEGMENT, 66
  SEGORDER, 74
  SET, 46
  SLT, 72
  SMODULES (global), 30
  SMODULES (segment), 67
  SOFTREF, 108
  specification order, 21
  STACK, 88
  START, 113
  SYMBOLS, 50
  SYSTEM, 114
  TITLE, 38
  TRIAL, 35
  TSTACK, 89
  UNSAT, 46
  USX, 40
  XFER, 44
  ZEROCOM, 116, 117
  ZSYMS, 115
Directives file, default, 17, 19
Directives processing order, 14
Directives, miscellaneous global, 50, 114
Directives, segment
  ENDSEG directive, 66
  segment description, 66
  SEGMENT directive, 66
  tree definition, 65
Directives, zero address, 115
Distributed EXpression table, definition, 150
Distributed mode, definition, 150
DUP directive
  example, 71, 125
  in nonsegmented loads, 75
  in segmented loads, 76, 77
  overview, 71
DUPENTRY directive
  default setting, 18
  keywords, 41
  overview, 41

Duplicate names
  entry point, 75
  module, 75
Duplicated
  common blocks and COMMONS directive, 76
  entry point errors, controlling, 32
Duplication
  definition, 75
  in nonsegmented loads, 75
    common blocks, 76
    entry point names, 75
    module names, 75
  in segmented loads, 76
    common blocks, 77
    entry point names, 77
    module names, 76
  program and block assignment, 75
  using DUP, 71
  using MODULES, 67
DUPLOAD directive, overview, 42
DUPORDER directive
  default setting, 18
  example, 33
  overview, 32
Dynamic common block
  allocation, 90
  and heap memory, 87
DYNAMIC directive
  example, 91, 124
  overview, 90
  with the heap, 92
Dynamic memory management, 87

E

ECHO directive
  as segment description directive, 66
  example, 128
  overview, 36
ENDSEG directive
  example, 67, 124, 125, 128
  overview, 66
ENDTREE directive
  example, 66, 121, 122, 124, 125, 128
  overview, 65
Entry point
  control and execution, 44, 113
  definition, 75, 150
  duplicate names, 75
  duplication
    error message control, 41, 42
    figure, 78
    in nonsegmented loads, 75
    in segmented loads, 77
    processing order, 32
  names, duplication in segmented loads, 75
  value assignment, 46
Entry point references
  testing with _loaded, 106
  testing with flag words, 107
Entry points, to external functions, 103
Environment variable processing, 12, 17
Environment variables
  LDDIR, 12
  LPP, 13
  MSG_FORMAT, 13
  NLSPATH, 13
  SEGDIR, 13
  TARGET, 14
  TMPDIR, 14
EQUIV directive
  example, 45
  overview, 45
Error, messages, 137
  descriptions, 138
  format, 139
  load–time, 137
  run–time, 137
Error messages
  controlling, directives for
    DUPENTRY, 41
    DUPLOAD, 42
    MLEVEL, 39
    MSGLEVEL, 43
    NODUPMSG, 42
    NOUSXMSG, 43
    REDEF, 40
    USX, 40
  overview, 39
  printing according to severity, 39
Events, definition, 150
Examples
  basic loader invocation, 121
  BIN directive, 26, 125
  BIN directive (segment), 70
  block maps, 131
  Comments, 37

common block reference map, 135
DEFDIR directive, 111
DEFLIB directive, 29
DUP directive, 71, 125
DUPORDER directive, 33
DYNAMIC directive, 91, 124
ECHO directive, 128
ENDSEG directive, 67, 124, 125, 128
ENDTREE directive, 66, 121, 122, 124, 125, 128
entry point cross-reference map, 134
entry point duplication, 78
EQUIV directive, 45
HEAP directive, 128
INCLUDE directive, 23
LIB directive, 27
LIBDIR directive, 111
LINCLUDE directive, 23
LLIB directive, 28
MAP directive, 128
map output, 130
MODULES directive, 31, 68, 124, 125, 129
MSGLEVEL directive, 43
NODEFLIB directive, 29
OMIT directive, 34
SAVE directive, 73
SEGMENT directive, 67, 124, 125, 128
segmented load with duplicated modules, 125
TREE directive, 66, 121, 122, 124, 125, 128
tree structure, 121, 122
figure, 123, 126
tree structure with expandable common block, 122
UNSAT directive, 46
Executable program, definition, 151
Executable program control, 34–35, 112
explain(1) command, 138
ext directive, 104
External functions, references to, 103
External reference, definition, 151
External symbols, 104

F

FATAL message, definition, 138
File attribute directive, OUTFORM, 112
File naming directives
ABS, 35
BIN, global, 26
LBIN, 26
LIB, 27
LLIB, 28
NODEFLIB, 28
Flag word usage, 107
Flag words, testing entry point references with, 107
FLOAT directive, 80
using automatic duplication, 80
using floating, 80
Floating, 80
definition, 151
FORCE directive
and data loads, 84
default setting, 18
overview, 30
Force-loading, definition, 151
Force-loading, definition, 30
Fortran program examples
acquiring space from the heap, 92
basic, 121
comprehensive
block maps, 131
common block reference map, 135
entry point cross-reference map, 134
Fortran program, 126
loader directives, 128
loader map output, 130
source code, 126
using dynamic common blocks, 93
FREEHEAP directive, 119

G

General directives, 21
Global directives
for segmentation, 72
miscellaneous
CASE, 51
COMPRESS, 52
CPUCHECK, 51
FLOAT, 80
INCFILE, 114
LOGFILE, 53
LOGUSE, 53
ORDER, 97
overview, 50, 114
SYMBOLS, 50
SYSTEM, 114
ZSYMS, 115

segmentation
COPY, 73
overview, 72
SAVE, 72
SEGORDER, 74
SLT, 72
Global heap memory, managing, 87, 117
Global segmentation directives, 72
COPY, 73
SAVE, 72
SEGORDER, 74
SLT, 72
Global symbol table, definition, 151

H

Hard references, converting to, 107
HARDREF directive, overview, 108
Header
creating, 38
example, 38
HEAP directive
example, 128
overview, 87
with DEFSTACK directive, 118
with DYNAMIC directive, 91
with STACK directive, 88
Heap memory
and Dynamic Common Block, 90
and the stack, 88, 118
global managing, 87, 117
Heap memory management directives
ADDBSS, 90
DEFHEAP, 117
DEFSTACK, 118
FREEHEAP, 119
HEAP, 87
STACK, 88
TSTACK, 89

I

INCFILE directive, overview, 114
Include, definition, 151
INCLUDE directive
example, 23
overview, 23
Including directives files, 22–23
Including object modules, 24–34
Initial transfer address, definition, 151
Initialization & alignment, program, 47
Initializing data areas, 47
Introduction, 1
Invoking SEGLDR, 3
ld command line, 10
segldr command line, 4

L

Languages supported, 1
LBIN directive, overview, 26
ld(1), command line, 10
LDDIR environment variable, 12
lib and bin files, exceptions and differences, 24
LIB directive
and BIN files, 27
example, 27
overview, 27
with DEFLIB directive, 29
with DUPORDER directive, 32
with FORCE directive, 30
with NODEFLIB directive, 28
LIBDIR directive
example, 111
overview, 111
Library, definition, 152
LINCLUDE directive
example, 23
overview, 23
LLIB directive
example, 28
overview, 28
Load map control, 35–38
Load–time messages, 137
Loader
definition, 152
invocation, 3
Loader directives, 15
conventions, 21
specification order, 21
Loader messages, 137
Loader resident routine ($SEGRES), 85
Loader-created tables, overview, 141
Loading
control of, 37

  - program segments, 1
- LOGFILE directive, overview, 53
- LOGUSE directive, overview, 53
- LPP environment variable, 13

M

- Machine characteristic checking, controlling, 51
- Magic number, definition, 152
- Managing global heap memory directives
  - ADDBSS, 90
  - DEFHEAP, 117
  - DEFSTACK, 118
  - FREEHEAP, 119
  - HEAP, 87
  - overview, 87, 117
  - STACK, 88
  - TSTACK, 89
- Map control, load, 37
- MAP directive
  - example, 128
  - overview, 37
- Map output
  - example, 130
  - generation, 37
- Mass storage, writing segment state to, 70, 72
- Memory, overlays, 1
- Message
  - descriptions, 138
  - format, 139
  - log file, format, 54
  - numbers, 138
- Messages
  - controlling error, 39
  - directives for
    - DUPENTRY, 41
    - DUPLOAD, 42
    - MLEVEL, 39
    - MSGLEVEL, 43
    - NODUPMSG, 42
    - NOUSXMSG, 43
    - REDEF, 40
    - USX, 40
- Miscellaneous global directives, 50, 114
- MLEVEL directive, overview, 39
- Module
  - assigning to segments, 66
  - definition, 152
  - duplication
    - in nonsegmented loads, 75
    - in segmented loads, 76
    - using DUP, 71
    - using MODULES, 67
    - using SMODULES, 67
  - names, duplication in nonsegmented loads, 75
  - names, duplication in segmented loads, 76
  - naming directives
    - FORCE, 30
    - MODULE (segment), 67
    - MODULES, (global), 30
    - SMODULE (segment), 67
    - SMODULES, (global), 30
  - zero address, 115
- MODULES directive
  - and data loads, 84
  - example, 124, 125
  - in segmented loads, 76
- MODULES directive (global)
  - example, 30
  - overview, 30
- MODULES directive (segment)
  - example, 68
  - overview, 67
- Movable block, definition, 152
- MSG_FORMAT, 139
- MSG_FORMAT environment variable, 13
- MSGLEVEL directive
  - example, 43
  - overview, 43

N

- Naming files
  - ABS directive, 35
  - BIN directive, (global), 26
  - LBIN directive, (global), 26
  - LIB directive, 27
  - LLIB directive, 28
  - NODEFLIB directive, 28
- NLSPATH environment variable, 13
- NODEFLIB directive
  - example, 29
  - overview, 28
- NODUPMSG directive, overview, 42
- Nonsegmented program, definition, 1
- NOTE message, definition, 137

NOUSXMSG directive, overview, 43

O

Object module
  definition, 152
  including, 34
OMIT directive
  example, 34
  overview, 34
ORDER directive, overview, 97
Ordered duplicate selection, definition, 152
ORG directive, overview, 49
OUTFORM directive, overview, 112
Output file attributes, OUTFORM directive, 112

P

Parallel virtual machine, definition, 153
Partition, definition, 153
Pascal, 1
PRESET directive, overview, 48
Primary entry point, definition, 153
Program alignment and initialization, 47–49
  directives
    ALIGN, 47
    ORG, 49
    PRESET, 48
  overview, 47
Program duplication & block assignment, 75
Program execution, segmented, 85
Program segmentation, introduction, 55
Program segments, loading, 1
Pseudo instructions and shared-text memory allocation, 98
PVM, definition, 153

R

REDEF directive, overview, 40
Relocatable binary module, definition, 153
Restrictions, data load, 83
Root segment, definition, 153
Root segment, definition, 55
Routines, block data, 84
Run-time messages, 137

S

SAVE directive
  example, 73
  with COPY directive, 73
SAVE directive (global), overview, 72
SAVE directive (segment), overview, 70
SCOMMONS directive (global), overview, 31
SCOMMONS directive (segment), overview, 68
Scratch file
  controlling position of, 85
  execution from, 73
SDT, description, 147
SDT, definition, 153
SECTION pseudo instructions, and shared-text memory allocation, 98
Sector, definition, 153
SEGDIR environment variable, 13
SEGLDR, invocation statements, 3
segldr(1), command line, 4
Segment
  convention, 1
  definition, 154
  definition directives, 65–66
  description directives, 66–71
  description, termination, 65
  linkage table (SLT), 72
  naming, 66
  predecessor, function, 55
  root, definition, 55
  subroutine calling between, 59
  successor, function, 55
  tree
    definition directives, 65
    design and restrictions, 57
    figures, 56, 57, 58, 59, 61, 62
    structure, 55
Segment branch, definition, 56
Segment description directives
  BIN, 69
  COMMONS, 68
  DUP, 71
  ENDSEG, 66
  MODULES, 67
  SAVE, 70
  SCOMMONS, 68
  SEGMENT, 66
  SMODULES, 67
  ZEROCOM, 116, 117

Segment Description table, definition, 153
Segment Description Table (SDT), description, 147
SEGMENT directive
  and DUP directive, 71
  example, 67, 124, 125, 128
  overview, 66
Segment Linkage Table, definition, 154
Segment Linkage Table (SLT), 72
  description, 146
Segment tree
  definition directives, 65–66
  restrictions, 57
Segment tree concept, 55
Segment tree definition, 55
Segment tree design, 57
Segmentation
  features, 55
  global directives, 72
Segmentation tables
  $SEGRES, 145
  overview, 144
  SDT, 147
  SLT, 146
Segmented load
  and COMMONS directive, 31
  and SCOMMONS directive, 31
  program, definition, 1
  with duplicated modules, 125
    example, 125
Segmented program, definition, 1
SEGORDER directive, overview, 74
SET directive, overview, 46
SHARED allocation scheme, overview, 98
Shared text program, 97
  creation, 97
  generation, 6
Shared-text allocation scheme
  advantages, 99
  disadvantages, 99
SLT, description, 146
SLT, definition, 154
SLT directive, overview, 72
SMODULES directive (global), overview, 30
SMODULES directive (segment), overview, 67
Soft externals
  how to convert to, 108
  how to convert to hard references, 107
  how to declare, 104–105
  how to link, 105
  overview, 103
  references, 103
  usage, 106–107
  usage (figure), 104
Soft references, 103
Soft references, converting to, 108
SOFTREF directive, overview, 108
Special purpose program, definition, 154
Stack and heap memory, 87, 117
STACK directive, 88
START directive
  command line equivalent for, 10, 16
  overview, 113
Subroutine
  call overhead, overview, 86
  calling between segments, 59
  illegal references, diagram, 61
SUMMARY message, definition, 138
Supported languages, 1
Symbolic debugging directives
  overview, 50, 114
  SYMBOLS, 50
SYMBOLS directive, overview, 50
SYSTEM directive, overview, 114
System libraries, default, 18

T

Tables
  $SEGRES, 145
  _infoblk, 141
  Segment Description Table (SDT), 147
  Segment Linkage Table (SLT), 146
TARGET environment variable, 14
Termination, of segment description, 65
TEXT, DATA, BSS, allocation scheme for memory allocation, 98
TITLE directive
  as segment description directive, 66
  example, 38
  overview, 38
TMPDIR environment variable, 14
Transfer entry point, definition, 154
TREE directive, example, 66, 121, 122, 124, 125, 128
TREE segment definition directive
  example, 66
  overview, 65

Tree structure
- basic example, 121
- figure, 126
- figures, 121, 122
- with expandable common block example, 123

Tree structure, example (figure), 123
Tree trimming, definition, 154
TRIAL directive, overview, 35
TSTACK directive, 89

U

UNICOS
- environment variable processing, 12
- ld(1) command line, 10
- segldr(1) command line, 4

UNSAT directive
- example, 46
- overview, 46

Unsatisfied external reference, definition, 154
Unsatisfied external references, 105
USX directive
- default setting, 18
- overview, 40

W

WARNING message, definition, 138

X

XFER directive
- command line equivalent for, 4, 15
- overview, 44

Z

Zero address, description directives, 115–117
ZEROCOM directive, overview, 116, 117
ZSYMS directive, overview, 115

Reader's Comment Form

Segment Loader (SEGLDR) and `ld` *Reference Manual* SR–0066 9.0

Your reactions to this manual will help us provide you with better documentation. Please take a moment to complete the following items, and use the blank space for additional comments.

List the operating systems and programming languages you have used and the years of experience with each.

Your experience with Cray Research computer systems: _____0-1 year _____1-5 year _____5+years

How did you use this manual: _____in a class _____as a tutorial or introduction _____as a procedural guide _____as a reference _____for troubleshooting _____other

Please rate this manual on the following criteria:

	Excellent			Poor
Accuracy	4	3	2	1
Appropriateness (correct technical level)	4	3	2	1
Accessibility (ease of finding information)	4	3	2	1
Physical qualities (binding, printing, illustrations)	4	3	2	1
Terminology (correct, consistent, and clear)	4	3	2	1
Number of examples	4	3	2	1
Quality of examples	4	3	2	1
Index	4	3	2	1

Please use the space below for your comments about this manual. Please include general comments about the usefulness of this manual. If you have discovered inaccuracies or omissions, please specify the number of the page on which the problem occurred.

Name ______________________

Title ______________________

Company ______________________

Telephone ______________________

Today's date ______________________

Address ______________________

City ______________________

State/Country ______________________

Zip code ______________________

Electronic mail address ______________________

Cut along this line

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY RESEARCH, INC.

ATTN: Software Publications Group
655 LONE OAK DR BLDG F
EAGAN MN 55121-9957

Fold

Reader's Comment Form

Segment Loader (SEGLDR) and `ld` *Reference Manual* SR–0066 9.0

Your reactions to this manual will help us provide you with better documentation. Please take a moment to complete the following items, and use the blank space for additional comments.

List the operating systems and programming languages you have used and the years of experience with each.

Your experience with Cray Research computer systems: _____0-1 year _____1-5 year _____5+years

How did you use this manual: _____in a class _____as a tutorial or introduction _____as a procedural guide _____as a reference _____for troubleshooting _____other

Please rate this manual on the following criteria:

	Excellent			Poor
Accuracy	4	3	2	1
Appropriateness (correct technical level)	4	3	2	1
Accessibility (ease of finding information)	4	3	2	1
Physical qualities (binding, printing, illustrations)	4	3	2	1
Terminology (correct, consistent, and clear)	4	3	2	1
Number of examples	4	3	2	1
Quality of examples	4	3	2	1
Index	4	3	2	1

Please use the space below for your comments about this manual. Please include general comments about the usefulness of this manual. If you have discovered inaccuracies or omissions, please specify the number of the page on which the problem occurred.

Name ______________________

Title ______________________

Company ______________________

Telephone ______________________

Today's date ______________________

Address ______________________

City ______________________

State/Country ______________________

Zip code ______________________

Electronic mail address ______________________

Cut along this line

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY
RESEARCH, INC.

ATTN: Software Publications Group
655 LONE OAK DR BLDG F
EAGAN MN 55121-9957

Fold