

Update Package

UPD4031-13A

for

PL/I Subset G Reference Guide
IDR4031

December 1984

This Update Package, UPD4031-13A, is Update 3 for the April 1980 edition of the PL/I Subset G, IDR4031. Pages that have been changed or added are listed on the next page.

A vertical bar in the margin of the table of contents indicates an addition since the last printing.

Copyright © 1984 by Prime Computer, Incorporated
Technical Publications Department
500 Old Connecticut Path
Framingham, MA 01701

The information contained on these updated pages is subject to change without notice and should not be construed as a commitment by Prime Computer Corporation. Prime Computer Corporation assumes no responsibility for any errors that may appear in this package.

PRIME and PRIMOS are registered trademarks of Prime Computer, Inc.

PRIMENET, RINGNET, Prime INFORMATION, PRIMACS, MIDASPLUS, Electronic Design Management System, EDMS, PDMS, PRIMEWAY, Prime Producer 100, INFO/BASIC, FW200, PST100, 2250, 9950, and THE PROGRAMMER'S COMPANION are trademarks of Prime Computer, Inc.

Update Package, UPD4031-13A

Pages to add:

<u>Insert Pages</u>	<u>After Page</u>
G-1 through G-14	F-8

Pages to change:

<u>Insert Pages</u>	<u>In Place of Pages</u>
i through viii	i through vi
X-1 through X-22	X-1 through X-20

PL/I Subset G Reference Guide

IDR4031

First Edition

Updated for Revision 19.4

by

Robert L. Dimmig

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 19.4 (Rev. 19.4).

**Prime Computer, Inc.
500 Old Connecticut Path
Framingham, Massachusetts 01701**

COPYRIGHT INFORMATION

The information in this document is subject to change without notice and should not be construed as a commitment by Prime Computer Corporation. Prime Computer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Copyright © 1984 by
Prime Computer, Incorporated
500 Old Connecticut Path
Framingham, Massachusetts 01701

PRIME and PRIMOS are registered trademarks of Prime Computer, Inc.

PRIMENET, RINGNET, Prime INFORMATION, PRIMACS, MIDASPLUS, Electronic Design Management System, EDMS, PDMS, PRIMEWAY, Prime Producer 100, INFO/BASIC, PST 100, PW200, PW150, 2250, 9950, THE PROGRAMMER'S COMPANION, and PRISAM are trademarks of Prime Computer, Inc.

PRINTING HISTORY — PL/I Subset G Reference Guide

<u>Edition</u>	<u>Date</u>	<u>Number</u>	<u>Software Release</u>
First Edition	April 1980	IDR4031	17.0
Update 1	December 1980	PIU2600-075	18.0
Update 2	July 1982	PIU2600-08	19.0
Update 3	December 1984	UPD4031-13A	19.4

Changes made to the text since the last printing are identified by vertical bars in the margin.

CUSTOMER SUPPORT CENTER

Prime provides the following toll-free numbers for customers in the United States needing service:

1-800-322-2838	(within Massachusetts)
1-800-541-8888	(within Alaska)
1-800-651-1313	(within Hawaii)
1-800-343-2320	(within other states)

HOW TO ORDER TECHNICAL DOCUMENTS

Obtain an order form, a catalog, and a price list from one of the following:

U.S. Customers

Software Distribution
Prime Computer, Inc.
1 New York Ave.
Framingham, MA 01701
(617) 879-2960 X2053

Prime Employees

Communications Services
MS 15-13, Prime Park
Natick, MA 01760
(617) 655-8000 X4837

Customers Outside U.S.

Contact your local Prime
subsidiary or distributor.

CONTENTS

PART I INTRODUCTION

1 INTRODUCTION

Definitions	1-1
This Document	1-1
Related Documents	1-2
The PL/I Language	1-3
The PL/I Subset G Language	1-3
The PLIG Language	1-4
Restrictions on PLIG Programs	1-5
Interface to Other Languages	1-6
PLIG and the Editor	1-7
PLIG and Prime Utilities	1-7
The Source Level Debugger	1-8
The PRIMOS Condition-Handling Mechanism	1-9
Conventions Used in this Guide	1-10

PART II THE PLIG LANGUAGE

2 OVERVIEW OF PLIG

Program Text	2-1
Names, Constants, and Punctuation	2-1
Comments	2-4
Text Replacement and Insertion	2-4
Statements	2-5
Declarations and References	2-8
Procedures	2-9
Block Structure and Scope	2-10
Parameters and Arguments	2-10
Block Activation and Recursion	2-12
Variables and Storage	2-13
Begin Blocks	2-15
Exception Handling	2-15
Input and Output	2-16

3 DATA AND DATA TYPES

Data Types	3-1
Arithmetic Types	3-1
Pictured Data	3-4
Character-String Data	3-7
Bit-String Data	3-8
Pointer Data	3-10
Label Data	3-12
Entry Data	3-14
File Data	3-17
Arrays	3-18

Structures 3-19
Arrays of Structure 3-21

4 STORAGE CLASSES

Storage Classes 4-1
Automatic Storage 4-1
Static Storage 4-2
Based Storage 4-3
Defined Storage 4-5
Parameters 4-6
Storage Sharing 4-8

5 DECLARATIONS AND ATTRIBUTES

Declarations 5-1
Label Prefixes 5-1
Declare Statements 5-3
Declaration Defaults 5-5
Attribute Consistency 5-6
Attributes 5-7

6 REFERENCES

Definitions 6-1
Simple and Subscripted References 6-1
Structure Qualified References 6-2
Pointer Qualified References 6-3
Procedure References 6-3
Built-In Function References 6-4
Variable References 6-4
Reference Resolution 6-5

7 EXPRESSIONS

Definition of Expression 7-1
Arithmetic Expressions 7-3
Relational Expressions 7-6
Bit-String Expressions 7-7
Concatenate Expressions 7-8

8 DATA TYPE CONVERSIONS

Introduction 8-1
Arithmetic to Arithmetic Conversion 8-2
Arithmetic to Bit-String Conversion 8-4
Arithmetic to Character-String Conversion 8-5
Bit-String to Arithmetic Conversion 8-6
Bit-String to Character-String Conversion 8-7
Character-String to Arithmetic Conversion 8-7
Character-String to Bit-String Conversion 8-8
Format Controlled Conversion 8-8
Pictured to Arithmetic Conversion 8-13
Pictured to Bit-String Conversion 8-13

Pictured to Character-String Conversion 8-14
Conversion to Pictured Data 8-14

9 STATEMENTS

How to Read this Section 9-1

10 BUILT-IN FUNCTIONS

Summary 10-1
Function Descriptions 10-1

PART III PL1G AND THE PRIME SYSTEM

11 IMPLEMENTATION DEFINED FEATURES

Arithmetic Precision 11-1
Maximum Sizes 11-1
Data Size and Alignment 11-2
Input/Output on TTY 11-2
Read and Write on Stream Files 11-3
Variable Length Input Lines 11-3
The Title Option and File Opening 11-3
Listing Control 11-5
Pointer Size Control 11-5
Additional Implementation Defined Features 11-5
Null Built-In Function 11-6
File System Limitation 11-6

12 ADVICE ON THE USE OF PL1G

Purpose of this Section 12-1
Efficiency 12-1
Common Programming Errors 12-2
Programming Style 12-3

13 PL1G USE OF THE CONDITION MECHANISM

Information Structure 13-1

14 USING THE PL1G COMPILER

Introduction 14-1
Invoking the Compiler 14-1
Compiler Error Messages 14-1
End-Of-Compilation Message 14-2
Compiler Options 14-3

APPENDICES

A GLOSSARY OF PL/I TERMS A-1

B ABBREVIATIONS B-1

C DATA FORMATS

- Overview C-1
- Fixed Binary Data C-2
- Fixed Decimal Data C-3
- Float Binary Data C-4
- Float Decimal Data C-5
- Picture Data D-6
- Character Data D-7
- Character Varying Data C-8
- Bit Data C-9
- Pointer Data C-10
- Label Data C-11
- Entry Data C-12
- File Data C-13

D STACK FRAME AND FUNCTION RETURN CONVENTIONS

- Locations of Returned Functions Values D-1
- Stack Frame Format D-1

E ASCII CHARACTER SET

- Prime Usage E-1
- Keyboard Input E-1
- Changing the Significance of Special Characters E-1

F DIFFERENCES BETWEEN FULL PL/I AND PL/I SUBSET G

- PL/I Features Supported in PL/I Subset G F-1
- PL/I Features Not Supported in PL/I Subset G F-1

G UPDATE FOR REVISION 19.4

- Updated Compiler Options G-1
- New Functionality G-10
- BIND: A New Linking/Loading Utility G-10

APPENDIX G
UPDATE FOR REVISION 19.4

This document is the third update to the First Edition of the PL/I Subset G Reference Guide (IDR4031). It includes features released at Master Disk Revision 19.4 (Rev. 19.4) plus some documentation corrections.

This document should be inserted as the last appendix, Appendix G, in your book.

UPDATED COMPILER OPTIONS

The changes in compiler options for Rev. 19.4 include:

- New syntaxes
- New options that have been added to implement new functionality
- New abbreviations for most options

These changes were implemented to make Prime's translator products more standardized in their user interface, as well as among themselves and with other Prime software products.

New Syntaxes

New syntaxes have been introduced to replace the older forms of some existing options. For example, the new syntax `-OPTIMIZE [n]`, where `n` is the level of optimization to be performed, has replaced the the older syntax of the options `-OPTIMIZE` and `-NOOPTIMIZE`.

Forming Negated Options: The construction of negated options is now done by prefixing the name of the positive form with `-NO_` (for example, `-MAP` and `-No_MAP`). To form the abbreviated name of a negated option, prefix the positive abbreviated name with `-N` (in this case, `-MAP` and `-NMAP`).

Specification of Options on Command Line: Processing of compiler options previously allowed duplicate and conflicting options on the same command line (for example, specifying an option twice, or specifying `-PRODUCTION` and `-NOPRODUCTION` for the same compiler invocation). In these occurrences of conflicting options, the one that had been specified last on the command line was the option that took control. Duplicate and conflicting options are now considered obsolete, and may cause fatal errors in future software revisions.

New Options

At Rev. 19.4, a number of new compile-time options replace current compiler options. The compiler will issue warnings if you should use the outdated forms, and the obsolete forms may cause fatal errors in future software revisions.

The updated options are:

```
-32I
-64V
-ALLOW_PRECONNECTION/-NO_ALLOW_PRECONNECTION
-BIG/-NO_BIG
-BINARY/-NO_BINARY
-COPY/-NO_COPY
-DEBUG/-NO_DEBUG
-ERRLIST/-NO_ERRLIST
-ERRITY/-NO_ERRITY
-EXPLIST/-NO_EXPLIST
-FRN/-NO_FRN
-FULL_HELP
-FULL_OPTIMIZE
-HELP
-INPUT pathname
-LCASE/-UPCASE
-LISTING [file]/-NO_LISTING
```

Updated options (continued)

-MAP/-NO_MAP
 -NESTING/-NO_NESTING
 -OFFSET/-NO_OFFSET
 -OPTIMIZE [n]
 -OVERFLOW/-NO_OVERFLOW
 -PRODUCTION/-NO_PRODUCTION
 -RANGE/-NO_RANGE
 -SILENT (with decimal argument)
 -SOURCE pathname
 -SPACE and -TIME
 -STATISTICS/-NO_STATISTICS
 -STORE_OWNER_FIELD/-NO_STORE_OWNER_FIELD
 -XREF/-NO_XREF

Following is a brief description of each option. The full name is listed along with the name of its opposite, negated form, if it has one. Proper abbreviations are indicated. Abbreviations that are now considered obsolete have been listed as such.

-32I

The -32I option generates 32I-mode code. There is no abbreviation for this option. Obsolete forms: -3, -32.

-64V

The -64V option generates 64V-mode code. There is no abbreviation for this option. Obsolete forms: -6, -64.

-ALLOW_PRECONNECTION/-NO_ALLOW_PRECONNECTION

You may specify the -ALLOW_PRECONNECTION option to permit the preconnection of a listing output to a pre-opened file unit 2, or a binary output to a pre-opened file unit 3. These preconnections must be made with the PRIMOS commands LISTING and BINARY, which are discussed in the PRIMOS Commands Reference Guide. If you specify the negative form of the compiler option, the compiler will always open and close the listing and binary files (and will use dynamic file units). The negative form is the default. Accepted abbreviations: -APRE for -ALLOW_PRECONNECTION, and -NAPRE for -NO_ALLOW_PRECONNECTION.

-BIG/-NO_BIG

The -BIG option can be used to handle arrays that span segment boundaries, if the compiler itself does not detect that these segment boundaries have been spanned. The negated option, -NO_BIG, is the default. Accepted abbreviations: -NBIG for -NO_BIG, no abbreviation for -BIG. Obsolete forms: -NOB, -NOBI, -NOBIG.

-BINARY [file]/-NO_BINARY

The **-BINARY** option specifies that a binary object file will be produced. The **-NO_BINARY** option replaces **-BINARY NO**, which is considered obsolete, to specify that no binary file is to be produced. The default is to produce a binary file. Accepted abbreviation: **-B [file]** for **-BINARY [file]** and **-NB** for **-NO_BINARY**. Obsolete forms: **-BI, -BIN, -BINA, -BINAR, -NOB, -NOBI, -NOBIN, -NOBINA, -NOBINARY**.

-COPY/-NO_COPY

To prevent constants from being changed by a called procedure, the **-COPY** option will make a copy of the constant before it is passed. The **-NO_COPY** option specifies that such a copy need not be made, on the assumption that the programmer has been careful not to pass constants as actual parameters to formals that get modified. **-COPY** is the default. Accepted abbreviations: **-COP** for **-COPY**, **-NCOP** for **-NO_COPY**.

-DEBUG/-NO_DEBUG

The **-DEBUG** option generates the full debugger (DBG) functionality code. Accepted abbreviations: **DBG, NDBG**. Obsolete forms: **-DE, -DEB, -DEBU, -NOD, -NODE, -NODEB, -NODEBU, -NODEBUG**.

-ERRLIST/-NO_ERRLIST

The **-ERRLIST** option prints a listing of errors only. Accepted abbreviations: **-ERRL** for **-ERRLIST**, **-NERRL** for **-NO_ERRLIST**. Obsolete forms: **-ERRLI, -ERRLIS, -NOERRL, -NOERRLI, -NOERRLIS, -NOERRLIST**.

-ERRTTY/-NO_ERRTTY

The **-ERRTTY** option prints error messages at user terminal. Accepted abbreviations: **-ERRT** for **-ERRTTY**, **-NERRT** for **-NO_ERRTTY**. Obsolete forms: **-ERRTT, -NOE, -NOERRT, -NOERRTT, -NOT, -NOTT, -NOTTY, -NOTTYD, -NOTTYDI, -NOTTYDIA, -NOTTYDIAG, -NOTTYDIAGS**.

-EXPLIST/-NO_EXPLIST

The **-EXPLIST** option prints a listing, including assembler-like output. Accepted abbreviations: **-EXP** for **-EXPLIST**, **-NEXP** for **-NO_EXPLIST**. Obsolete forms: **-EX, -EXPL, -EXPLI, -EXPLIS, -NOEX, -NOEXP, -NOEXPL, -NOEXPLI, -NOEXPLIS**.

-FRN/-NO_FRN

The **-FRN** option generates special code to improve the accuracy of single-precision floating-point calculations (by doing a floating round on store instructions). Accepted abbreviations: **-NFRN** for **-NO_FRN**. Obsolete forms: **-F, -FR, -NOF, -NOFR, -NOFRN**.

-FULL_HELP

This option is similar to the **-HELP** option, except that in addition to the usage summary, a description of each compiler option is given. Accepted abbreviation: **-FH**.

-FULL_OPTIMIZE

This option indicates that the highest possible optimization should be used. The program listing will display the optimization level raised by the use of this option. Accepted abbreviation: **-FOPT**.

-HELP

The **-HELP** option will produce information on how to use the invoked compiler, including a list of compiler options. However, descriptions of the options are not provided. The user is referred to the system **HELP** command to obtain full information about the installed compiler, and to the **-FULL_HELP** option for full information about the invoked compiler. Accepted abbreviation: **-H**.

Note: If the user makes a mistake in specifying the options, or gives only the name of the compiler on the command line, then the user will be referred to the **-HELP** option after the error message has been given.

-INPUT pathname

This is an alternate way of specifying the source of the input file. If **pathname** is **TTY**, then input will come from the user terminal. Accepted abbreviation: **-I** **pathname**. Obsolete forms: **-IN**, **-INP**, **-INPU**.

-LCASE/-UPCASE

These compiler options control the compiler's automatic conversion to uppercase. When **-UPCASE** is used, everything in the source program, except literals, is translated to uppercase; when **-LCASE** is used, the source program (except for reserved words) is left as is. With **-LCASE** a program can have two data items (**X** and **x**, for example) which are treated as separate items. The **-UPCASE** option is the default. Accepted abbreviations: **-UP** for **-UPCASE**, **-LC** for **-LCASE**. Obsolete forms: **-LCA**, **-LCAS**, **-U**, **-UPC**, **-UPCA**, **-UPCAS**.

-LISTING [file]/**-NO_LISTING**

The **-LISTING** option is used both to specify that a source file listing is to be produced and, optionally, to specify where the listing is to go. The **-NO_LISTING** option takes the place of **-LISTING NO**, which is considered obsolete, to specify that no listing file is to be produced. The negative option is the default. Accepted abbreviations: **-L** for **LISTING** [file], **-NL** for **-NO_LISTING**. Obsolete forms: **-LI**, **-LIS**, **-LIST**, **-LISTI**, **-LISTIN**.

-MAP/-NO_MAP

The **-MAP** option produces information similar to the **-XREF** option, but without the source line references. When you specify **-MAP**, you will automatically have a listing file produced. Conversely, when you specify **-L**, you will automatically get the map information. If

you do not want your listing file to have the map information, you need to specify -L with -NO_MAP, as -MAP is the default. Accepted abbreviations: -MA for -MAP, -NMA for -NO_MAP. Obsolete form: -NO_M.

-NESTING/-NO_NESTING

The -NESTING option adds nesting level numbers in the program listing. Accepted abbreviations: -NE for -NESTING, -NNE for -NO_NESTING. Obsolete forms: -NONE, -NONES, -NONEST, -NONESTI, -NONESTIN, -NONESTING.

-OFFSET/-NO_OFFSET

The -OFFSET option appends an offset map to the listing file. Its use also implies that -L must be specified on the command line. For each statement in the source program, the offset map gives the offset in the object file of the first machine instruction generated for that statement. The -NO_OFFSET option means that no offset map is created, and is the default. Accepted abbreviations: -OFF for -OFFSET, -NOFF for -NO_OFFSET. Obsolete forms: -O, -OF, -OFFS, -OFFSE, -NOOF, -NOOFF, -NOOFFS, -NOOFFSE, -NOOFFSET.

-OPTIMIZE [dec-integer]

This option is the same as the existing optimize option except that it may be followed by a decimal integer that specifies an optimization level. If the decimal integer is not specified, the default value is 2. To turn optimization off, -OPT 0 should be used. -NO_OPTIMIZE IS now considered to be obsolete. Accepted abbreviation: -OPT [dec-integer]. Obsolete forms: -OP, -OPT1, -OPT2, -OPT3, -NOOP, -NOOPTIMIZE.

Following is a brief description of what types of optimizations are performed at each level. Each optimization level performs all the optimizations of the next lower level, plus those that are listed. Note that the functionality associated with some levels at Revision 19.4 may change in the future, and that more levels of optimization may be added at later revisions.

- 0: Perform no optimizations. This level replaces the option -NOOPTIMIZE.
- 1: Pattern replacement.
- 2: Common subexpression elimination.
- 3: Loop invariant removal.

-OVERFLOW/-NO_OVERFLOW

The -OVERFLOW option enables integer overflow conditions. The negated option, -NO_OVERFLOW, is the default. Accepted abbreviations: -OVF for -OVERFLOW, -NOVF for -NO_OVERFLOW. Obsolete forms: -OVE, -OVER, -OVERF, -OVERFL, -OVERFLO, -NOOV, -NOOVE, -NOOVER, -NOOVERF, -NOOVERFL, -NOOVERFLO, -NOOVERFLOW.

-PRODUCTION/-NO_PRODUCTION

These options are alternative options controlling code for Prime's Source Level Debugger, DBG. The **-PRODUCTION** option is similar to **-DEBUG**, except that no statement information is produced. With **-PRODUCTION**, you may not breakpoint at individual statements, nor step through your program. The execution time of a program compiled with **-PRODUCTION** will be better than the execution time of a program compiled with **-DEBUG**. The **-NO_PRODUCTION** option is the default. Accepted abbreviations: **-PROD** for **-PRODUCTION**, and **-NPROD** for **-NO_PRODUCTION**. Obsolete forms: **-P**, **-PR**, **-PRO**, **-PRODU**, **-PRODUC**, **-PRODUCT**, **-PRODUCTI**, **-PRODUCTIO**, **-NOP**, **-NOPR**, **-NOPRO**, **-NOPROD**, **-NOPRODU**, **-NOPRODUC**, **-NOPRODUCT**, **-NOPRODUCTI**, **-NOPRODUCTIO**, **-NOPRODUCTION**.

-RANGE/-NO_RANGE

These compiler options control runtime subscript range checking. When **-RANGE** is specified on the compile line, any variable array subscript is checked for out-of-range conditions, and if any occurs, a runtime error is issued and the program halts. The use of **-RANGE** will increase both the compile time and execution time of a program, and therefore should be used only as a debugging tool. The **-NO_RANGE** option will suppress this capability, and is the default option. Accepted abbreviations: **-RA** for **-RANGE**, and **-NRA** for **-NO_RANGE**. Obsolete forms: **-R**, **-RAN**, **-RANG**, **-NOR**, **-NORA**, **-NORAN**, **-NORANG**, **-NORANGE**.

-SILENT [n]

This option has been modified to accept a decimal argument, which indicates an error severity level. When **n** is specified, all errors with that severity and less are NOT reported. The default value of **n** is 1. Accepted abbreviation: **-SI**. Obsolete forms: **-SIL**, **-SILL**, **-SIL2**, **-SIL3**, **-SILE**, **-SILEN**, **-SILENT**, **-SILENT1**, **-SILENT2**, **-SILENT3**.

-SOURCE pathname

This is an alternate way of specifying the source of the input file. If **pathname** is **TTY**, then input will come from the user terminal. Accepted abbreviation: **-S pathname**. Obsolete forms: **-SO**, **-SOU**, **-SOUR**, **-SOURC**.

-SPACE and -TIME

The **-SPACE** option specifies that the size of the optimized code (space) is to be given preference over the speed of the optimized code (time) in optimization consideration. The opposite of **-SPACE** is **-TIME**, which means that optimization is to favor time over space. The **-TIME** option is the default. There are no abbreviations for either of these two options.

-STATISTICS/-NO_STATISTICS

The **-STATISTICS** option displays compilation statistics at the terminal. The **-NO_STATISTICS** option suppresses compile time statistics, which may be generated by the **-STATISTICS** option. The negative option is the default. Accepted abbreviations: **-STAT** for **-STATISTICS**, **-NSTAT** for **-NO_STATISTICS**. Obsolete forms: **-STAT1**,

-STATIS, -STATIST, -STATISTI, -STATISTIC, -NOST, -NOSTA, -NOSTAT,
 -NOSTATI, -NOSTATIS, -NOSTATIST, -NOSTATISTI, -NOSTATISTIC,
 -NOSTATISTICS, -TO, -TOT, -TOTA, -TOTAL, -TOTALS.

-STORE_OWNER_FIELD/-NO_STORE_OWNER_FIELD

The `-STORE_OWNER_FIELD` option causes the identity of the current program to be stored in a known place for use by traceback routines, such as DMSTK. This option is the default. Using the `-NO_STORE_OWNER_FIELD` option will not save this information, but will save a small code sequence for extremely time-critical programs. Accepted abbreviations: `-SOF` for `-STORE_OWNER_FIELD`, and `-NSOF` for `-NO_STORE_OWNER_FIELD`. Obsolete forms: `-STO`, `-STOR`, `-STORE`, `-STORE_`, `-STORE_O`, `-STORE_OW`, `-STORE_OWN`, `-STORE_OWNE`, `-STORE_OWNER`, `-STORE_OWNER_`, `-STORE_OWNER_F`, `-STORE_OWNER_FI`, `-STORE_OWNER_FIE`, `-STORE_OWNER_FIEL`, `-NOSTO`, `-NO_STOR`, `-NO_STORE`, `-NO_STORE_`, `-NO_STORE_O`, `-NO_STORE_OW`, `-NO_STORE_OWN`, `-NO_STORE_OWNE`, `-NO_STORE_OWNER`, `-NO_STORE_OWNER_`, `-NO_STORE_OWNER_F`, `-NO_STORE_OWNER_FI`, `-NO_STORE_OWNER_FIE`, `-NO_STORE_OWNER_FIEL`, `-NSO`, `-NO_OW`, `-NO_OWN`, `-NO_OWNE`, `-NO_OWNER`, `-NO_OWNERI`, `-NO_OWNERID`.

-XREF/-NO_XREF

The `-XREF` option will produce a listing with cross references of data and procedure names. Accepted abbreviations: `-XR` for `-XREF`, `-NXR` for `-NO_XREF`. Obsolete forms: `-X`, `-XRE`, `-NOX`, `-NOXRE`, `-NOXREF`.

New Abbreviations

A number of abbreviations for compiler options have been modified. The old abbreviations will still be accepted at Revision 19.4, but will not be accepted at later revisions.

Following is a list of PL1G compiler options, and their official abbreviations. Default options are preceded by an asterisk (*).

<u>Option</u>	<u>Abbreviation</u>
-32I	(none)
* -64V	(none)
-ALLOW_PRECONNECTION	-APRE
* -NO_ALLOW_PRECONNECTION	-NAPRE
-BIG	-(none)
* -NO_BIG	-NBIG
* -BINARY [file]	-B [file]
-NO_BINARY	-NB
* -COPY	-COP
-NO_COPY	-NCOP
-DEBUG	-DBG
* -NO_DEBUG	-NDBG
-ERRLIST	-ERRL
* -NO_ERRLIST	-NERRL
* -ERRITY	-ERRT
-NO_ERRITY	-NERRT
-EXPLIST	-EXP
* -NO_EXPLIST	-NOEXP
-FRN	-FRN
* -NO_FRN	-NFRN
-FULL_HELP	-FH
-FULL_OPTIMIZE	-FOPT
-HELP	-H
-INPUT pathname	-I pathname
-LCASE	-LC
-LISTING [file]	-L [file]
* -NO_LISTING	-NL
* -MAP	-MA
-NO_MAP	-NMA
-NESTING	-NE
* -NO_NESTING	-NNE
-OFFSET	-OFF
* -NO_OFFSET	-NOFF
* -OPTIMIZE [dec-integer]	-OPT [dec-integer]
-OVERFLOW	-OVF
* -NO_OVERFLOW	-NOVF
-PRODUCTION	-PROD
* -NO_PRODUCTION	-NPROD
-RANGE	-RA
* -NO_RANGE	-NRA
* -SILENT [dec-integer]	-SI [dec-integer]
-SOURCE pathname	-S pathname
-SPACE	(none)
-STATISTICS	-STAT

<u>Option (continued)</u>	<u>Abbreviation</u>
* -NO_STATISTICS	-NSTAT
* -STORE_OWNER_FIELD	-SOF
-NO_STORE_OWNER_FIELD	-NSOF
* -TIME	(none)
* -UPCASE	-UP
-XREF	-XR
* -NO_XREF	-NXREF

NEW FUNCTIONALITY

Locked MIDAS + Records

The ON ERROR statement can now be used to catch the raised condition when a locked record of a MIDAS file is referenced. The corresponding value returned by the built-in function ONCODE is 1130, and it can be used to distinguish the special condition from other error conditions.

Installation and Build Procedures

PL1G is now supplied with two directories: PL1G and PL1G_LIBRARY. In order to use PL1G, the user must run two install files, PL1G>PL1G.INSTALL.COM1 and PL1G_LIBRARY>PL1G_LIBRARY.INSTALL.COM1.

Obsolete Options

This is the last revision that will accept the -OBDATA and -NOOBDATA options. They are the same as the -EXPLIST and -NO_EXPLIST options, which are the more acceptable forms.

BIND: A NEW LINKING/LOADING UTILITY

An important new feature you should know about at Rev. 19.4 is Prime's new, more efficient program loading utility named BIND. If you are using a Rev. 19.4 or higher system, it would be advantageous for you to use this new loader. Prime's former loader, SEG, is still available to all users. However, in this update and future revisions of this book, BIND, not SEG, will be documented and used in examples. If you want more information on SEG, you should consult the SEG and LOAD Reference Guide.

Complete information on Prime's new loader, BIND, is contained in the new Programmer's Guide to BIND and EPFs. This book is also released at Rev. 19.4. In this update, however, a minimum amount of BIND information will be presented to get you started.

What is BIND?

BIND is Prime's new dynamic linking/loading utility for loading V-mode and I-mode programs. The executable files (runfiles) that BIND creates from the object (binary) files produced by Prime's high-level languages are called EPFs (Executable Program Formats).

Used for all virtual addressing mode programs, BIND makes it easier for you to build and maintain software. Unlike SEG, BIND creates dynamic EPF runfiles. This means that these runfiles can execute in any segment, or segments, of PRIMOS and do not need to use the same segment each time they are invoked. These runfiles can be assigned by PRIMOS at runtime, instead of load time, into free segments of your address space. A runfile can also be interrupted, and then resumed with the PRIMOS RESUME command.

SEG, on the other hand, creates static runfiles. That is, with SEG, all segment locations used by a program are assigned at load time by PRIMOS, and the runfile uses the same segment every time it executes. Whereas EPFs can be suspended and restarted at the PRIMOS level without intervening programs overwriting and destroying the suspended program, SEG's runfiles cannot.

One of BIND's most convenient features is that you can actually create an EPF on one PRIMOS command line. That is, the entire loading process can be entered on just one line.

Another advantage of BIND is the ability to invoke certain programs via DBG, without causing the program and Debugger to compete for the same storage space. As described in Appendix M of the Source Level Debugger User's Guide (UPD4033-21A), when SEG and DBG are called upon to process programs that use MIDAS+ or DBMS, the resulting storage overlap may lead to runtime errors. Although it is possible to reload the program and use A/SYMBOL commands to sidestep the problem, BIND's dynamic EPF runfiles avoid this problem altogether.

Using BIND

After compiling your program in the usual way, with the suffix .PL1G appended to the source file, you can use BIND in one of two ways:

- Directly on one PRIMOS command line
- Interactively by invoking subcommands of BIND

Invoking on One Command Line: To run BIND from the PRIMOS command line, enter the command:

```
BIND [epf-filename] -LO[object-file...] -LI PL1GLB -LI
```

When using BIND on a PRIMOS command line, you must precede each argument (BIND command) with a hyphen (-). The epf-filename is the name of your runfile (EPF). This name need not be the name of your source program. If the epf-filename is missing, BIND uses the name of the first loaded object file.

A loading session using one PRIMOS command line looks like this:

```
OK, BIND TEST -LOAD TEST TEST.2 -LIBRARY PL1GLB -LIBRARY
[BIND rev 19.4]
BIND COMPLETE
OK,
```

In the example shown above, the -LOAD argument is used to load the main program, TEST, and any other binary file, in this case, TEST.2. The -LIBRARY PL1GLB argument loads the PL1G library, and the -LIBRARY command loads the system library.

When the bind is complete, BIND saves your runfile (EPF), with the default name epf-filename.RUN, in the directory to which you are attached. If you do not specify an epf-filename, BIND automatically adds the suffix .RUN to the first object file that you load and saves the runfile in your directory. In the example above, the name of the runfile is:

```
TEST.RUN
```

Invoking Interactively: To invoke BIND interactively, enter:

```
BIND
```

at PRIMOS command level. BIND then asks you with a colon (:) prompt to load your files and libraries. An interactive loading session looks like this:

```
OK, BIND TEST
[BIND rev 19.4]
: LOAD TEST
: LOAD TEST.2
: LIBRARY PL1GLB
: LIBRARY
BIND COMPLETE
: FILE
OK,
```

BIND creates your runfile in the directory to which you are attached and names it TEST.RUN. The FILE command saves the EPF runfile and returns you to PRIMOS command level.

Executing Runfiles Created with BIND: To execute a runfile created with BIND, enter the command:

RESUME filename

Other commands: Three other useful BIND commands are QUIT, MAP, and HELP. The QUIT command returns you to PRIMOS immediately without saving the current EPF. Using the MAP command with the -UNDEFINED option identifies unresolved references if you do not receive a "BIND COMPLETE" message. HELP can give you online help on the commands if you run into trouble while using BIND. For more information see the Programmer's Guide to BIND and EPFs.

Where to Substitute BIND for SEG

In the PL/I Subset G Reference Guide, there is one reference to SEG. To update your book, you should note the following page, where SEG is referenced, then substitute BIND information in place of SEG. The SEG reference appears on the following page:

1-3

Index

Numbers

- 32I compiler option, 14-9, G-2, G-3
- 64V compiler option, 14-9, G-2, G-3

A

A format, 8-11

Abbreviations,

- For built-in functions, B-1
- For compiler options, 14-10
- For keywords, B-1
- Updated (Rev. 19.4) list of, G-9
- Use of, 12-6

ABS built-in function, 10-1

ACOS built-in function, 10-2

Addition, 7-4

ADDR built-in function, 3-12, 10-2

Addressing modes, 14-9

Adjustable,

- Arrays, 3-18
- Extents, 4-6
- Parameters, 4-6

Advice on PL1G programming, 12-1

Aggregates,

- And BIG compiler option, 14-9

ALIGNED attribute, 3-9, 5-7

Alignment,

- Of data types, 11-2

ALLOCATE statement, 4-3, 9-1

ALLOW_PRECONNECTION compiler option, G-2, G-3

Alternate data descriptions, 4-5

Ambiguous references, 6-2

ANSI standards, 1-1

APPEND files, 11-4

- Argument lists,
 - In general, 9-27
- Arguments,
 - And shared storage, 4-9
 - Expressions as, 4-6
 - In general, 2-10, 4-6, 6-3
 - In procedure calls, 9-6
 - Match with parameters, 5-11
 - Matched with parameters, 4-7
 - Mismatched with parameters, 4-6, 9-6
 - Passed by-reference, 4-6
 - Passed by-value, 4-6
 - Type-conversion of, 9-6
- Arithmetic,
 - Conversion precisions for, 8-3
 - Expressions, 7-3
 - Operands, 7-3
 - Operators, 7-3
 - Precision, 11-1
- Arithmetic Data, 3-1
- Array data, 3-18
- Arrays,
 - Adjustable, 3-18
 - As parameters, 4-7
 - Assignment to, 3-19, 9-2
 - Declaring, 3-18, 5-4, 5-9
 - Extents of, 4-1
 - I/O of, 3-19, 9-19, 9-30
 - In general, 3-18
 - Initialization of, 5-12
 - Maximum no. of dimensions, 3-18
 - Of labels, 3-14, 5-2
 - Of structures, 3-21
 - Processed as wholes, 3-19
 - Range checking, 3-19
 - Referencing, 3-19, 4-4, 6-1
 - Shared storage and, 4-8
 - Size limit on internal, 11-1
 - Storage order, 3-18
 - Subscripts of, 3-19
 - Within structures, 3-21
- ASCII character set, 11-6, E-1
- ASIN built-in function, 10-2
- Assignment,
 - And references, 9-2
 - In list-directed input, 9-20
 - Rules for, 9-2
 - Statements, 9-1
 - To arrays, 3-19, 9-2
 - To bit data, 3-9
 - To bit-string data, 9-2
 - To character data, 3-8, 9-2
 - To entry data, 3-15
 - To file data, 3-17
 - To fixed-point data, 3-3
 - To label data, 3-12
 - To picture data, 3-6
 - To pointer data, 3-10
 - To structures, 3-20, 9-2
- Assignments,
 - Using PAGENO, 9-3
 - Using STRING, 9-3
 - Using SUBSTR, 9-4
 - Using UNSPEC, 9-4
- ATAN built-in function, 10-2
- ATAND built-in function, 10-2
- ATANH built-in function, 10-3
- Attributes,
 - ALIGNED, 3-9, 5-7
 - All for data types, A-8
 - AUTOMATIC, 5-7
 - BASED, 4-8, 5-8
 - BINARY, 3-2, 3-3, 5-8
 - BIT, 3-9, 5-8
 - BUILTIN, 5-8
 - Changing, 2-20
 - CHARACTER, 3-7, 5-9
 - Check for validity, 2-19
 - Consistency of, 5-6
 - DECIMAL, 3-2, 3-3, 5-9
 - Default, 2-19
 - Default in declarations, 5-5
 - DEFINED, 4-8, 5-9
 - DIMENSION, 5-9
 - DIRECT, 2-19, 2-24, 5-10
 - Duplicate, 5-5
 - ENTRY, 3-1, 3-14, 5-10
 - EXTERNAL, 4-2, 5-1, 5-11
 - FILE, 2-17, 3-17, 5-11
 - FIXED, 3-2, 5-11
 - FLOAT, 3-3, 5-12
 - For DECLARE statement, 5-7

- Attributes (continued)
 Implied, 2-19
 In general, 2-19, 5-7
 In OPEN statement, 9-26
 Incomplete, 9-26
 INITIAL, 5-12
 INPUT, 2-19, 2-24, 5-13
 INTERNAL, 4-2, 5-13
 KEYED, 2-19, 2-24, 5-13
 LABEL, 3-12, 5-13
 Legal combinations of, 9-26
 Of files, 2-19
 OUTPUT, 2-19, 2-24, 5-13
 PICTURE, 3-4, 5-14
 POINTER, 3-10, 5-14
 PRINT, 2-19, 2-21, 5-14, 11-5
 RECORD, 2-19, 5-14, 9-26
 Required, 2-19
 RETURN, 3-15
 RETURNS, 5-15
 SEQUENTIAL, 2-19, 2-24, 5-15
 STATIC, 5-15
 STREAM, 2-19, 5-15, 9-26
 UPDATE, 2-19, 2-24, 5-15
 VARIABLE, 3-15, 5-10, 5-16
 VARYING, 3-8, 5-16
- AUTOMATIC attribute, 5-7
- AUTOMATIC storage class, 2-14,
 4-1
- B
- B format, 8-12
- Base of arithmetic data, 3-1
- BASED attribute, 4-8, 5-8
- BASED storage class, 4-3
- Based variables, 12-1
- BEGIN blocks, 2-15, 5-2, 9-5,
 9-12
- BEGIN statement, 9-5
- BIG compiler option, 14-9, G-2,
 G-3
- BINARY attribute, 3-2, 3-3, 5-8
- BINARY built-in function, 10-3
- BINARY compiler option, 14-8,
 G-2, G-4
- Binary notation, 2-3
- BIND,
 background, G-11
 examples of use, G-12
 executable program formats
 (EPF's), G-11
- BIT attribute, 3-9, 5-8
- BIT built-in function, 10-3
- BIT data, 3-9, C-9
- Bit-string,
 Expressions, 7-7
 Operands, 7-8
 Operators, 7-7
 Type-conversions, 7-8
- Blanks,
 Blank lines, 12-5
 For I/O control, 11-3
 In list-directed I/O, 2-22
 Required, 2-4
 Significance of, 2-3
 Trailing, 11-3
- Blocks,
 Activation of, 2-12
 All types defined, A-4
 And GO TO statement, 9-22
 And procedures, 2-9
 And reference resolution, 6-5
 BEGIN, 2-15, 5-2, 9-5, 9-12
 File control, 2-17
 Inactive, 3-13, 3-15
 Structure of, 2-10
- BOOL built-in function, 10-3
- Boolean values, 3-9
- Built-in function,
 ONLOC, 13-1

Built-in functions,

ABS, 10-1
 ACOS, 10-2
 ADDR, 3-12, 10-2
 ASIN, 10-2
 ATAN, 10-2
 ATAND, 10-2
 ATANH, 10-3
 BINARY, 10-3
 BIT, 10-3
 BOOL, 10-3
 BYTE, 10-4, 11-6
 CEIL, 10-4
 CHARACTER, 10-4
 COLLATE, 10-4, 11-1
 COPY, 10-5
 COS, 10-5
 COSD, 10-5
 COSH, 10-5
 DATE, 10-5
 DECIMAL, 10-5
 Declaring, 6-4
 DIMENSION, 10-6
 DIVIDE, 10-6
 EXP, 10-6
 FIXED, 10-6
 FLOAT, 10-7
 FLOOR, 10-7
 HBOUND, 10-7
 In general, 10-1
 INDEX, 10-8
 LBOUND, 10-8
 LENGTH, 10-9
 LINENO, 2-21, 10-9
 Listed, 10-1
 LOG, 10-9
 LOG10, 10-9
 LOG2, 10-10
 MAX, 10-10
 MIN, 10-10
 MOD, 10-10
 NULL, 3-12, 10-10, 11-6
 ONCODE, 2-16, 10-11, 13-1
 ONFILE, 2-17, 10-11, 13-1
 ONKEY, 10-11, 13-1
 ONLOC, 10-11
 PAGENO, 2-21, 10-11
 Pointer valued, 6-3
 Precision of, 11-6
 RANK, 10-11, 11-6
 Referencing, 6-4
 ROUND, 10-12
 SIGN, 10-12
 SIN, 10-12

Built-in functions (continued)

SIND, 10-12
 SINH, 10-12
 SQRT, 10-12
 STRING, 10-12
 SUBSTR, 10-13, 12-2
 TAN, 10-13
 TAND, 10-13
 TANH, 10-13
 TIME, 10-13
 TRANSLATE, 10-13
 TRUNC, 10-14
 UNSPEC, 10-14
 VALID, 10-14
 VERIFY, 10-15

BUILTIN attribute, 5-8

By-reference, passing, 4-6

By-value, passing, 4-6

BYTE built-in function, 10-4,
11-6

C

CALL statement, 9-6

Calls,

In general, 2-11

To procedures, 5-10, 9-27

Case conversion, 14-5

Case statement (simulated),
3-14, 5-2

CEIL built-in function, 10-4

Changing attributes, 2-20

CHARACTER attribute, 3-7, 5-9

CHARACTER built-in function,
10-4

CHARACTER data, 3-7, C-7

Character set, ASCII, 11-6, E-1

CHARACTER VARYING data, 3-7, C-8

- Character-string, length of, 3-7
- CLOSE statement, 2-20, 9-6
- COL format, 9-14
- COLLATE built-in function, 10-4, 11-1
- Comments, 12-2
 - In general, 2-4
 - Run-on, 2-4
- Comparison,
 - Of bit-strings, 3-9
- Comparisons,
 - Of character strings, 3-8
 - Of entry data, 3-15
 - Relational, 7-7
- Compiler, (See also Options)
 - End-of-compilation message, 14-2
 - Error messages, 14-1
 - In general, 14-1
 - Invoking, 14-1
 - Options, 14-1
 - Severity code, 14-2
- Compiler options,
 - 32I, 14-9, G-3
 - 64V, 14-9, G-3
 - Abbreviations for, 14-10
 - ALLOW_PRECONNECTION, G-3
 - BIG, 14-9, G-3
 - BINARY, 14-8, G-4
 - COPY, G-4
 - DEBUG, 14-9, G-4
 - ERRLIST, G-4
 - ERRITY, G-4
 - EXPLIST, 14-6, G-4
 - FRN, G-4
 - FULL_HELP, G-5
 - FULL_OPTIMIZE, G-5
 - HELP, G-5
 - In general, 14-3
 - INPUT, 14-5
 - INPUT pathname, G-5
 - LCASE, 14-5, G-5
 - LISTING, 14-6, G-5
 - MAP, G-5
 - NESTING, 14-7, G-6
 - NO_ALLOW_PRECONNECTION, G-3
 - NO_BIG, G-3
 - NO_BINARY, G-4
 - NO_COPY, G-4
 - NO_DEBUG, G-4
 - NO_ERRLIST, G-4
 - NO_ERRITY, G-4
 - NO_EXPLIST, G-4
 - NO_FRN, G-4
 - NO_LISTING, G-5
 - NO_MAP, G-5
 - NO_NESTING, G-6
 - NO_OFFSET, G-6
 - NO_OVERFLOW, G-6
 - NO_PRODUCTION, G-7
 - NO_RANGE, G-7
 - NO_STATISTICS, G-7
 - NO_STORE_OWNER_FIELD, G-8
 - NO_XREF, G-8
 - OFFSET, 14-7, G-6
 - OPTIMIZE, 14-10
 - OPTIMIZE [n], G-6
 - OVERFLOW, G-6
 - PRODUCTION, 14-10, G-7
 - RANGE, 12-2, 14-10, G-7
 - SILENT, 14-7
 - SILENT (with decimal argument), G-7
 - SOURCE, 14-5, G-7
 - SPACE, G-7
 - STATISTICS, 14-7, G-7
 - STORE_OWNER_FIELD, G-8
 - Table of, 14-4, 14-11
 - TIME, G-7
 - UPCASE, 14-5, G-5
 - Updated List of, G-2
 - XREF, 14-6, G-8
- Compound statements, 2-6
- Concatenation expressions, 7-8
- Concatenation operand, 7-8
- Concordance, 14-6
- Condition handler, 1-9, 9-24, 13-1
- Condition handling, 2-15
- Conditions, 1-9
 - And condition handler, 13-1
 - And ON statement, 9-23

Conditions (continued)

ENDFILE, 9-24
 ENDPAGE, 2-16, 2-21, 9-25
 ERROR, 2-16, 9-25
 In general, 2-16
 KEY, 9-25
 SIGNAL statement for, 9-35
 Signalling, 9-24

Constants,

All types defined, A-7
 Bit, 3-10
 Character string, 3-8
 File, 2-17, 3-17
 Fixed point, 3-3
 Floating point, 3-4
 In general, 2-2
 Label, 3-12

Control block, file, 2-17

Control characters, restricted,
 2-21

Control formats for I/O, 2-23

Conventions, 1-10

COPY built-in function, 10-5

COPY compiler option, G-2, G-4

COS built-in function, 10-5

COSD built-in function, 10-5

COSH built-in function, 10-5

Creation of files, 2-24

Cross reference, 14-6

D

DAM files, 11-4

Data,

Alignment of, 11-2
 Arithmetic, 3-1
 Arrays, 3-18
 Arrays of structures, 3-20
 Automatic, 2-14, 4-1

Data (continued)

Base of arithmetic, 3-1
 Based, 4-3
 Constant size, 12-1
 Defined (storage class), 4-5
 Fixed-point, 3-2
 Floating-point, 3-3
 Formats for, 9-14, C-1
 In general, 2-13, 3-1
 Operands, 7-3
 Packed decimal, 3-2
 Parameters, 4-6
 Precision of arithmetic, 3-1
 Relational operands, 7-7
 Rounding arithmetic, 3-3
 Scale of arithmetic, 3-1
 Static, 2-14, 4-2
 Structures, 3-19
 Structures of arrays, 3-20
 Type-conversion of, 8-1
 Undefined, 6-5
 Varying size, 12-1

Data types,

BIT, 3-9

Data formats, 9-14

Data types,

And prefix operators, 7-4
 BIT, C-9
 CHARACTER, 3-7, C-7
 CHARACTER VARYING, 3-7, C-8
 Conversion of, 7-3
 Efficiencies of using, 12-1
 ENTRY, 3-14, C-12
 FILE, 3-17, C-13
 FIXED BINARY, 3-1, C-2
 FIXED DECIMAL, 3-1, C-3
 FLOAT BINARY, 3-1, C-4
 FLOAT DECIMAL, 3-1, C-5
 For operands, 7-3
 For relational operands, 7-7
 Formats for, C-1
 In general, 2-13, 3-1
 LABEL, 3-12, C-11
 Mismatched in expressions, 7-3
 Of constants, 3-1
 Of expression results, 3-1
 Picture, 8-14
 PICTURE, 3-4, C-6
 POINTER, 3-10, C-10
 Size of, 11-2
 Valid, listed, 5-6

- Data-types,
 - Converting, 8-1
- Database Management System (DBMS), 1-7
- DATE built-in function, 10-5
- DBMS, 1-7
- DEBUG compiler option, 14-9, G-2, G-4
- Debugger, 1-9, 14-9
- Debugging, 14-10
- DECIMAL attribute, 3-2, 3-3, 5-9
- DECIMAL built-in function, 10-5
- Declarations,
 - DECLARE statement, 9-7
 - Defaults for, 5-5
 - Defaults for invalid, 5-6
 - Duplicate attributes in, 5-5
 - ENTRY without VARIABLE, 5-7
 - Errors in, 5-6
 - Factored, 5-5
 - FILE without VARIABLE, 5-7
 - In general, 2-8, 5-1, 5-3
 - Incomplete, 5-6
 - Inconsistent, 5-6
 - Invalid, 5-6
 - Multiple, 5-1
 - Of arrays, 3-18, 5-4, 5-9
 - Of arrays of structures, 3-21
 - Of bit data, 3-9
 - Of built-in functions, 6-4
 - Of character data, 3-7
 - Of defined data, 4-5
 - Of entry data, 3-14
 - Of external procedures, 3-15
 - Of file data, 3-17
 - Of fixed-point data, 3-2
 - Of floating-point data, 3-3
 - Of label data, 3-12, 3-14
 - Of labels, 2-8, 5-1, 5-3
 - Of parameters, 9-27
 - Of picture data, 3-5
 - Of pointer data, 3-10
 - Of procedure names, 5-3
 - Of static data, 2-14
 - Of structures, 3-20, 5-4
- Declarations (continued)
 - Of structures of arrays, 3-21
 - Of variables, 5-3
 - Recommended forms, 5-3
 - Redeclarations, 6-2
 - Redundant, 5-1
 - Scope of, 5-1
- DECLARE statement, 2-8, 2-19, 5-3, 5-7, 9-7
- Defaults,
 - Attribute, 2-19
 - For binary data, 5-8
 - For decimal data, 5-9
 - For declarations, 5-5
 - For EXTERNAL attribute, 5-11
 - For file names, 11-3
 - For file-creation, 2-24
 - For FIXED attribute, 5-11, 5-12
 - For FLOAT attribute, 5-12
 - For GET statement, 9-18
 - For implicit OPEN, 2-19
 - For invalid declarations, 5-6
 - For line size, 2-19
 - For LINESIZE option, 11-1
 - For opening files, 11-5
 - For opening nonexistent files, 2-24
 - For PAGESIZE option, 11-1
 - For PUT statement, 9-29
 - For STATIC attribute, 5-13
 - For static storage scope, 4-2
 - For TITLE, 2-19
 - On-unit, 1-10
 - Storage class, 4-1
 - System on-unit, 1-10
- DEFINED attribute, 4-8, 5-9
- DEFINED storage class, 4-5
- DELETE statement, 2-19, 2-23, 9-7
- Deletion of files, 2-24
- DEVICE files, 11-4
- Device names, 11-4
- DIMENSION attribute, 5-9

DIMENSION built-in function, 10-6

Dimensions of an array, 3-18

DIRECT attribute, 2-19, 2-24, 5-10

DIVIDE built-in function, 10-6

Division, 7-5

DO REPEAT statement, 9-9

DO statement (iterative), 9-10

DO statement (simple), 9-7

DO WHILE statement, 9-8

Duplicate attributes, 5-5

E

E format, 8-10

EDIT option, 9-21, 9-32

Edit-directed I/O,
 Efficiency of, 12-1
 In general, 2-20, 2-23

Edit-directed input,
 In general, 9-21

Edit-directed output,
 In general, 9-32

Editor, and PLLG, 1-7

Efficiency,
 And choice of data type, 12-1
 And edit-directed I/O, 12-1
 And keys, 12-1
 And list-directed I/O, 12-1
 And record I/O, 12-1
 In choice of data type, 3-2
 Of binary data, 12-1
 Of decimal data, 12-1
 With arrays, 12-1
 With based variables, 12-1
 With structures, 12-1

ELSE statement, 9-22

END statement, 9-11

End-of-compilation message, 14-2

ENDFILE condition, 9-24

ENDPAGE condition, 2-16, 2-21, 9-25

ENTRY attribute, 3-1, 3-14, 5-10

ENTRY data, 3-14, C-12

Entry point references, 6-4

Equality of data, 7-7

ERRLIST compiler option, G-2, G-4

ERROR condition, 2-16, 9-25

Errors,

 And on-units, 9-24

 Compiler error messages, 14-2

 Compiler-detected, 14-1

 Handling of, 2-15

 In declarations, 5-6

 In general, 12-2

 In type-conversion, 8-4

 Inconsistent procedure declarations, 12-2

 Invalid pointers, 12-2

 Invalid recursive invocations, 12-3

 Missing semicolons, 12-2

 Null pointers, 12-2

 Overflow values, 12-3

 RANGE compiler option, 12-2

 %%REPLACE of keywords, 12-2

 Run-on comments, 12-1

 Run-on string constants, 12-2

 Runtime (See The Prime User's Guide)

 SUBSTR built-in function, 12-2

 Undefined variables, 12-3

ERRITM compiler option, G-2, G-4

ESCAPE editor command, 1-7

- Evaluation,
 - Altering order of, 7-2
 - And parentheses, 7-2
 - Due to assignment, 9-2
 - Of expressions, 7-1
 - Of format-list, 9-19
 - Of IF expressions, 9-22
 - Of input-list, 9-19
 - Order of, 7-1, 9-3, 9-6
 - Partial, 7-2
 - Exception handling, 2-15
 - Exclusions, 1-5
 - Executing programs (See The Prime User's Guide)
 - Execution, order of, 2-7
 - EXP built-in function, 10-6
 - Expanded listing, 14-6
 - EXPLIST compiler option, 14-6, G-2, G-4
 - Exponent, 3-3
 - Exponentiation, 7-5
 - Exponents, 11-5
 - Expressions,
 - Altering evaluation of, 7-2
 - Arithmetic, 7-3
 - Bit-string, 7-7
 - Concatenation, 7-8
 - Data types from, 3-1
 - Defined, 7-1
 - Evaluated on assignment, 9-2
 - Evaluation of, 7-1
 - Expression operators, 7-1
 - For extents, 4-1
 - In general, 7-1
 - In IF statements, 9-22
 - In output lists, 9-30
 - Parentheses in, 7-2
 - Partial evaluation of, 7-2
 - Precision resulting from, 7-3
 - Relational, 7-6, 7-7
 - Subscript, 6-1
 - Extensions, 1-4
 - Extents,
 - Adjustable (*) parameter, 4-6
 - Defined, 4-1
 - For automatic data, 4-1
 - For static data, 4-2
 - Of arrays, 4-1
 - Of based variables, 4-5
 - Of defined data, 4-6
 - Of parameters, 4-6
 - Of variables, 4-1
 - Parameter, 4-7
 - EXTERNAL attribute, 4-2, 5-1, 5-11
- F
- F format, 8-9
 - Factored declarations, 5-5
 - FILE attribute, 2-17, 3-17, 5-11
 - FILE data, 3-17, C-13
 - FILE option, 2-18, 9-26
 - Files, (See also File Attributes)
 - All types defined, A-12
 - And TITLE option, 11-3
 - APPEND, 11-4
 - Attributes of, 2-19
 - Closing, 2-20, 9-6
 - Creation of, 2-24
 - DAM, 11-4
 - Deletion of, 2-24
 - Description formats for, 11-4
 - DEVICE, 11-4
 - Direct, 5-10
 - DIRECT, 11-5
 - File constants, 2-17
 - File control blocks, 2-17, 3-17
 - FILE option, 2-18
 - File variable, 2-17
 - In general, 2-17
 - INPUT, 5-13
 - KEYED, 5-13
 - Line size, 2-20
 - Names as parameters, 2-17
 - Names for, 11-3

Files (continued)

Opening, 2-18, 9-26, 11-3
 Opening nonexistent, 2-24
 Operations on record, 2-24
 OUTPUT, 5-13
 Position of, 2-23, 2-24
 PRINT, 5-14
 Properties of, 2-18
 Record, 2-17, 2-23
 RECORD, 5-14
 Replacement of, 2-24
 Restriction on accessing, 11-6
 SAM, 11-4
 SEQUENTIAL, 5-15
 Stream, 2-17, 2-20, 11-3
 STREAM, 5-15
 UPDATE, 5-15

FIXED attribute, 3-2, 5-11

FIXED BINARY data, 3-1, C-2

FIXED built-in function, 10-6

FIXED DECIMAL data, 3-1, C-3

Fixed-point constants, 3-3

Fixed-point data, 3-2

FLOAT attribute, 3-3, 5-12

FLOAT BINARY data, 3-1, C-4

FLOAT built-in function, 10-7

FLOAT DECIMAL data, 3-1, C-5

Floating point data, 3-3

FLOOR built-in function, 10-7

Format labels, 5-2

Format lists,

In general, 9-13

Items for, 9-13

Nested, 9-13

Repetition factors in, 9-13

Format names, 5-2

FORMAT statement, 9-12

Formats,

A, 8-11

B, 8-12

COL, 9-14

E, 8-10

F, 8-9

In general, 9-12

LINE, 9-15

P, 8-13

PAGE, 9-16

R, 9-14

SKIP, 9-16

TAB, 9-17

X, 9-17

Formats for data types, C-1

Formatting programs, 12-4

FORMS, 1-8

Forms Management System (FORMS),
1-8

Fractions, 3-2

FREE statement, 4-5, 9-17

FRN compiler option, G-2, G-4

FROM option, 2-24, 11-3

FULL_HELP compiler option, G-2,
G-5

FULL_OPTIMIZE compiler option,
G-2, G-5

Fully qualified references, 6-2

Function references, 6-3

Functions,

BUILTIN, 5-7, 5-8

Pointer valued, 6-3

Referencing, 6-3

Referencing built-in, 6-4

Return conventions for, D-1

G

GET statement, 2-19, 2-20, 9-18,
11-2, 11-3

Glossary, A-1

GO TO statement, 9-21

H

HBOUND built-in function, 10-7

HELP compiler option, G-2, G-5

Hexadecimal notation, 2-3

I

Identifiers, 2-2
(See also Names)

IF statement, 2-6, 9-22

Implementation-defined features,
1-5, 11-1

Implied attributes, 2-19

Implied-DO, 9-19, 9-31
 In input-lists, 9-19
 In output-lists, 9-29

INCLUDE statement, 2-4, 11-5

INDEX built-in function, 10-8

Inequality of data, 7-7

Infix operators, 7-1, 7-8

INITIAL attribute, 5-12

Initialization,
 Attribute for, 5-12
 In general, 2-14, 5-12
 Of arrays, 5-12
 Of static storage, 4-2

INPUT attribute, 2-19, 2-24,
5-13

INPUT compiler option, 14-5

INPUT pathname compiler option,
G-2, G-5

Input-lists,
 Implied-DO in, 9-19
 In general, 9-18

Input/Output,
 A format, 8-11
 And devices, 11-4
 And type-conversion, 8-8
 B format, 8-12
 CLOSE statement, 9-6
 COL format, 9-14
 Control formats, 2-23
 Data formats, 9-14
 DELETE statement, 9-7
 E format, 8-10
 Edit-directed, 2-20, 2-23
 Edit-directed input, 9-21
 Edit-directed output, 9-32
 Efficiency of, 12-1
 ENDFILE condition, 9-24
 ENDPAGE condition, 9-25
 F format, 8-9
 Format lists, 9-13
 FORMAT statement, 9-12
 Formats for, 9-13
 GET statement, 9-18
 Implied-DO, 9-19
 Implied-DO in, 9-31
 In general, 2-16
 Input lists, 9-18
 Input-line length, 11-3
 KEY condition, 9-25
 LINE format, 9-15
 Line size during, 2-20
 List-directed, 2-20, 2-22
 List-directed input, 9-20
 List-directed output, 9-31
 Of arrays, 3-19, 9-19, 9-30
 Of structures, 3-20, 9-19,
 9-30
 Of variable-length lines, 11-3
 On TTY, 11-2
 OPEN statement, 9-26
 P format, 8-13
 PAGE format, 9-16
 PUT statement, 9-29

- Input/Output (continued)
 - R format, 9-14
 - READ statement, 9-33
 - Record, 2-23, 12-1
 - REWRITE statement, 9-35
 - SKIP format, 9-16
 - Stream, 2-20
 - TAB format, 9-17
 - Transmitted string size, 11-2
 - WRITE statement, 9-36
 - X format, 9-17
- Insertion of text, 2-4
- Integers, 3-2
- Interfacing other languages, 1-6
- INTERNAL attribute, 4-2, 5-13
- INTIO option, 2-24, 11-3
- Invoking procedures, 2-11
- IO statement Output-lists, 9-29
- K
- KEY condition, 9-25
- KEYED attribute, 2-19, 2-24, 5-13
- Keys,
 - In record files, 2-24
 - Size-restriction on, 11-6
- KEYTO option, 11-6
- %%keyword (See the keyword)
- Keywords,
 - In general, 2-2
 - Reserved, 2-2
- L
- LABEL attribute, 3-12, 5-13
- LABEL data, 3-12, C-11
- Label prefixes (See Labels)
- Labels,
 - Arrays of, 3-14, 5-2
 - Declaration of, 5-3
 - Declaring, 2-8, 5-1
 - Effect of overuse, 2-7
 - On BEGIN blocks, 5-2
 - On FORMAT statement, 5-2
 - Statement, 5-2
 - Subscripted, 5-2
 - Subscripts of, 3-14
- IBOUND built-in function, 10-8
- LCASE compiler option, 14-5, G-2, G-5
- Left-to-right equivalent, 4-8
- Length,
 - Of a key, 11-6
 - Of bit data, 3-9
 - Of character data, 3-7
 - Of concatenated strings, 7-8
 - Of input lines, 11-3
 - Of names, 11-2
 - Of string constant, 11-1
 - Of string value, 11-1
- LENGTH built-in function, 10-9
- Lengths,
 - Mismatch during I/O, 2-23
- Level numbers, 5-4
- Level numbers in structures, 3-19
- Line boundaries, marking, 2-22
- LINE format, 9-15
- Line numbers, 2-21
- Line size during I/O, 2-20
- LINENO built-in function, 2-21, 10-9
- LINESIZE option, 2-19, 11-1

- Linking programs (See Loading programs)
 - LIST option, 9-20, 9-31
 - LIST statement, 11-5
 - List-directed I/O,
 - Efficiency of, 12-1
 - In general, 2-20, 2-22
 - List-directed input,
 - In general, 9-20
 - List-directed output,
 - In general, 9-31
 - LISTING compiler option, 14-6, G-2, G-5
 - Loading programs (See The Prime User's Guide)
 - LOG built-in function, 10-9
 - LOG10 built-in function, 10-9
 - LOG2 built-in function, 10-10
 - Lower-to-upper-case, 14-5
- M
- MAIN, 9-28
 - Major structures, 3-19
 - Mantissa, 3-3
 - MAP compiler option, G-2, G-5
 - Marking line boundaries, 2-22
 - Marking page boundaries, 2-22
 - Matching arguments with parameters, 4-7
 - MAX built-in function, 10-10
 - Members of structures, 3-19
 - Memory formats, C-1
 - MIDAS, 1-8
 - MIN built-in function, 10-10
 - Mismatch of arguments with parameters, 4-6
 - MOD built-in function, 10-10
 - Module, program, 2-1
 - Multiple declarations, 5-1
 - Multiple Index Data Access System (MIDAS), 1-8
 - Multiple qualification, 6-3
 - Multiplication, 7-4
- N
- Names,
 - Declared, 2-2
 - External, 11-2
 - File, 2-17
 - For devices, 11-4
 - For files, 11-3
 - In general, 2-1
 - Keyword, 2-2
 - Length of, 11-2
 - Of structure components, 3-20
 - Qualified, 3-20
 - Reserved, 2-2
 - Scope of, 2-10
 - Suggestions for selecting, 12-6
 - Syntax for, 2-1
 - Nested format lists, 9-13
 - NESTING compiler option, 14-7, G-3, G-6
 - Nesting level, 14-7
 - NO_ALLOW_PRECONNECTION compiler option, G-2, G-3
 - NO_BIG compiler option, G-2, G-3

- NO_BINARY compiler option, G-2, G-4
 - NO_COPY compiler option, G-2, G-4
 - NO_DEBUG compiler option, G-2, G-4
 - NO_ERRLIST compiler option, G-2, G-4
 - NO_ERRITY compiler option, G-2, G-4
 - NO_EXPLIST compiler option, G-2, G-4
 - NO_FRN compiler option, G-2, G-4
 - NO_LISTING compiler option, G-2, G-5
 - NO_MAP compiler option, G-2, G-5
 - NO_NESTING compiler option, G-3, G-6
 - NO_OFFSET compiler option, G-3, G-6
 - NO_OVERFLOW compiler option, G-3, G-6
 - NO_PRODUCTION compiler option, G-3, G-7
 - NO_RANGE compiler option, G-3, G-7
 - NO_STATISTICS compiler option, G-3, G-7
 - NO_STORE_OWNER_FIELD compiler option, G-3, G-8
 - NO_XREF compiler option, G-3, G-8
 - NOLIST statement, 11-5
 - Nonexistent files, 2-24
 - NULL built-in function, 3-12, 10-10, 11-6
 - Null pointer, 3-11
 - Null statement, 9-23
 - Null string, 3-7, 3-9
- O
- OBDATA/NOOBDATA (See EXPLIST/NO_EXPLIST)
 - OBDATA/NOOBDATA compiler option, G-10
 - Object code, 14-9
 - Object file, 14-8
 - Octal notation, 2-3
 - OFFSET compiler option, 14-7, G-3, G-6
 - Offset map, 14-7
 - ON ERROR statement, G-10
 - ON statement, 2-6, 2-15, 9-23
 - On-units, 1-9, 13-1
 - And GO TO statement, 9-24
 - And ON statement, 9-23
 - And RETURN statement, 9-34
 - In general, 2-16, 9-23
 - SIGNAL statement for, 9-35
 - ONCODE built-in function, 2-16, 10-11, 13-1
 - ONFILE built-in function, 2-17, 10-11, 13-1
 - ONKEY built-in function, 10-11
 - ONKEY built-in functions, 13-1
 - ONLOC built-in function, 10-11, 13-1

- OPEN statement, 2-18, 2-19, 9-26, 11-3
- Operands,
 Arithmetic, 7-3
 Bit-string, 7-8
 Concatenation, 7-8
 Data types of, 7-3
 Relational, 7-7
- Operators, 2-3
 All types defined, A-17
 Arithmetic, 7-3
 Bit-string, 7-7
 In expressions, 7-1
 In general, 7-1
 Infix, 7-1, 7-8
 Prefix, 7-1, 7-4
 Priority of, 7-1
 Relational, 7-6
 Results of, 7-3
- Optimization, 14-10
- OPTIMIZE [n] compiler option, G-3, G-6
- OPTIMIZE compiler option, 14-10
- Options, (See also Compiler options)
 EDIT, 9-21, 9-32
 FILE, 2-18, 9-26
 FROM, 2-24, 11-3
 INFO, 2-24, 11-3
 KEYTO, 11-6
 LINESIZE, 2-19, 11-1
 LIST, 9-20, 9-31
 PAGE, 2-21
 PAGESIZE, 2-19, 11-1
 RECURSIVE, 2-12, 9-27
 RETURNS, 9-27
 SKIP, 2-20
 TITLE, 2-17, 2-18, 9-26, 11-3
- OPTIONS (MAIN), 9-28
- Order of evaluation, 7-1, 9-3, 9-6
- Order of execution, 2-7
- Out-of-bounds values, 14-10
- OUTPUT attribute, 2-19, 2-24, 5-13
- Output lists,
 Implied-DO in, 9-29
- Output-lists,
 Implied-DO in, 9-31
 In general, 9-29
- OVERFLOW compiler option, G-3, G-6
- Overlays, string, 4-8
- Overview of PL1G, 2-1
- P
- P format, 8-13
- Packed decimal data, 3-2
- Page boundaries, marking, 2-22
- PAGE format, 9-16
- Page numbers, 2-21
- PAGE option, 2-21
- PAGENO built-in function, 2-21, 10-11
- PAGENO pseudo-variable, 9-3
- PAGESIZE option, 2-19, 11-1
- Parameter lists,
 In general, 9-27
- PARAMETER storage class, 4-6
- Parameters,
 And BEGIN blocks, 2-15
 And shared storage, 4-9
 Arrays as, 4-7
 Associated by-reference, 4-6
 Associated by-value, 4-6
 Extents of, 4-6
 File names as, 2-17
 In general, 2-10, 4-6

- Parameters (continued)
 - Match with arguments, 5-11
 - Matched with arguments, 4-7
 - Mismatched with arguments, 4-6
 - Structures as, 4-7, 5-11
 - With adjustable extents, 4-6
 - With constant extents, 4-7
 - With ENTRY attribute, 5-11
- Parentheses in expressions, 7-2
- Parity bit, E-3
- Partially qualified references, 6-2
- Passing by-reference, 4-6
- Passing by-value, 4-6
- PICTURE attribute, 3-4, 5-14
- Picture characters, 3-5
- Picture data, 8-14
- PICTURE data, 3-4, C-6
- PL/I Subset G, Defined, 1-1
- PL/I, defined, 1-1
- PLIG,
 - And other languages, 1-6
 - And Prime utilities, 1-7
 - And the condition handler, 1-9
 - And the debugger, 1-9
 - And the editor, 1-7
 - Defined, 1-1
 - Exclusions, 1-5
 - Extensions, 1-4
 - Restrictions on, 1-6
- PLIG,
 - Overview of, 2-1
- POINTER attribute, 3-10, 5-14
- POINTER data, 3-10, C-10
- Pointer qualified references, 6-3
- Pointers,
 - All types defined, A-18
 - And based data, 4-3
 - Explicit qualification of, 4-5
 - Implicit qualification of, 4-4
 - Invalid, 12-2
 - Null, 12-2
 - Qualification of, 4-3
 - Qualified, 12-6
 - Size control of, 11-5
- Precision,
 - After addition, 7-4
 - After division, 7-5
 - After exponentiation, 7-5
 - After multiplication, 7-4
 - After subtraction, 7-4
 - After type-conversion, 8-3
 - And expression evaluation, 7-4
 - And prefix operators, 7-4
 - Arithmetic, 11-1
 - Defined, A-18
 - Fixed binary, 11-1
 - Fixed decimal, 11-1
 - Float binary, 11-1
 - Float decimal, 11-1
 - In expressions, 7-3
 - Mismatch in expressions, 7-3
 - Of arguments, 4-7
 - of arithmetic data, 3-1
 - Of arithmetic data, 4-7
 - Of binary data, 5-8
 - Of built-in functions, 11-6
 - Of decimal data, 5-9
 - Of FIXED data, 5-12
 - Of fixed-point results, 7-3
 - Of FLOAT data, 5-12
 - Of floatint-point results, 7-4
 - Of parameters, 4-7
 - Rules for arithmetic, 7-3
- Precision of arithmetic constants, 3-3
- Prefix operators, 7-1
- PRINT attribute, 2-19, 2-21, 5-14, 11-5
- Priority of operators, 7-1
- Procedure references, 6-3
- PROCEDURE statement, 5-11, 9-27

- Procedures,
 All types defined, A-18
 And blocks, 2-9
 Calls to, 9-27
 Entry points of, 5-10
 External, 2-9
 In general, 2-9
 Inconsistent declaration of,
 12-2
 Internal, 2-9, 12-3
 Invoking, 2-11
 Nested, 2-9
 Programming style for, 12-3
 Recommendations for, 12-3
 Recursive, 2-12, 9-28, A-20
 References to, 9-6
 Scope of, 2-10
- PRODUCTION compiler option,
 14-10, G-3, G-7
- Program execution (See The Prime
 User's Guide)
- Program organization, 12-3
- Programming style, 12-3
- Programs,
 Defined, 2-1
 Execution order, 2-7
 From other systems, 2-1
 Modules in, 2-1
- Programs, formatting style for,
 12-4
- Pseudo-variables,
 PAGENO, 2-21, 9-3
 STRING, 9-3
 SUBSTR, 9-4
 UNSPEC, 9-4
- Punctuation, 2-3
- PUT statement, 2-19, 2-20, 9-29,
 11-2
- Q
- Qualification, multiple, 6-3
- Qualified references, 6-2
- Quotes, 2-3, 3-8
- R
- R format, 9-14
- Range checking, 14-10
- Range checking in arrays, 3-19
- RANGE compiler option, 14-10,
 G-3, G-7
- RANK built-in function, 10-11,
 11-6
- READ statement, 2-19, 2-23,
 9-33, 11-3
- RECORD attribute, 2-19, 5-14
- Record file, 2-17
- Record I/O, 2-23
- Record size, 11-4
- Recursion,
 And entry data, 3-16
 And static variables, 4-2
 In general, 2-12
 In procedures, 9-28
 Invalid recursive invocations,
 12-3
- RECURSIVE option, 2-12, 9-27
- Redundant declarations, 5-1
- References,
 Ambiguous, 6-2, 6-5
 And assignments, 9-2
 And redeclarations, 6-2
 Fully qualified, 6-2
 Function, 6-3
 In general, 2-8, 6-1
 Invalid, 6-5
 Partially qualified, 6-2
 Pointer qualified, 6-3
 Qualified, 6-2

References (continued)

- Resolving, 6-1, 6-5
 - Simple, 6-1, 6-5
 - Structure qualified, 6-2
 - Subscripted, 6-1, 6-2, 6-5
 - To arrays, 6-1
 - To BASED variables, 6-3
 - To built-in functions, 6-4, A-6
 - To entry points, 6-4
 - To functions, 6-3
 - To procedures, 6-3, 9-6
 - To structures, 6-2, 6-5
 - To undefined data, 6-4
 - To variables, 6-4
- Referencing, (See also
References)
- Arrays, 3-19
 - Arrays in structures, 3-21
 - Based data, 4-3
 - Pointer-qualified, 4-3
 - Shared storage, 4-9
 - Structures, 3-20, 12-6
 - Structures in arrays, 3-21
 - With pointers, 4-3
- Relational,
- Expressions, 7-6, 7-7
 - Operands, 7-7
 - Operators, 7-6
- Repetition factors, 9-13
- REPLACE statement, 12-2
- REPLACE Statement, 2-5
- Replacement of files, 2-24
- Replacement of text, 2-4
- Required attributes, 2-19
- Required blanks, 2-4
- Reserved keywords, 2-2
- Reserved names, 2-2
- Resolving references, 6-5
- Restrictions on PLIG programs,
1-6
- RETURN statement, 9-34
- Returns,
In general, 2-11
- RETURNS attribute, 3-15, 5-15
- RETURNS option, 9-27
- REVERT statement, 9-34
- REWRITE statement, 2-19, 2-23,
9-35
- ROUND built-in function, 10-12
- Rounding arithmetic data, 3-3
- Row-major order, 3-18
- Running programs (See The Prime
User's Guide)
- S
- SAM files, 11-4
- Scale factors, 11-1, A-21
- Scale of arithmetic data, 3-1
- Scope,
- Of BEGIN statement label, 5-2
 - Of declarations, 5-1
 - Of names, 2-10
 - Of overlapping structures, 6-2
 - Of procedures, 2-10
 - Of static data, 4-2
 - With EXTERNAL attribute, 5-11
 - With FILE attribute, 5-11
 - With INTERNAL attribute, 5-13
- SEMICO editor command, 1-7
- Separators, 2-3, A-21
- SEQUENTIAL attribute, 2-19,
2-24, 5-15
- SET clause, A-1
- Severity code, 14-2

- Sharing storage, 4-8
- SIGN built-in function, 10-12
- SIGNAL statement, 9-35
- Signalling conditions, 9-24
- SILENT (with decimal argument)
 compiler option, G-3, G-7
- SILENT compiler option, 14-7
- Simple references, 6-1, 6-5
- Simulated case statement, 5-2
- SIN built-in function, 10-12
- SIND built-in function, 10-12
- SINH built-in function, 10-12
- SKIP format, 9-16
- SKIP option, 2-20
- SOURCE compiler option, 14-5,
 G-3, G-7
- Source file, 14-5
- Source level debugger, 14-9
- Source listing, 11-5, 14-6
- Source text standards, 2-1
- Source-level debugger, 1-9
- SPACE compiler option, G-3, G-7
- SQRT built-in function, 10-12
- Stack frame, 2-12, 3-16, A-22
- Stack frame format, D-1
- Standards, 1-1
- Statement labels, 5-2
- Statements,
 - ALLOCATE, 4-3, 9-1
 - Assignment, 9-1
 - BEGIN, 9-5
 - CALL, 9-6
 - Case (simulated), 3-14
 - CLOSE, 2-20, 9-6
 - Compound, 2-6
 - DECLARE, 2-8, 2-19, 5-3, 5-7,
 9-7
 - DELETE, 2-19, 9-7
 - DO (iterative), 9-10
 - DO (simple), 9-7
 - DO REPEAT, 9-9
 - DO WHILE, 9-8
 - ELSE, 9-22
 - END, 9-11
 - FORMAT, 9-12
 - FREE, 4-5, 9-17
 - GET, 2-19, 2-20, 9-18, 11-2,
 11-3
 - GO TO, 9-21
 - IF, 2-6, 9-22
 - In general, 2-5, 9-1
 - INCLUDE, 2-4, 11-5
 - LIST, 11-5
 - NOLIST, 11-5
 - Null, 9-23
 - ON, 2-6, 2-15, 9-23
 - OPEN, 2-18, 2-19, 9-26, 11-3
 - Order of execution, 2-7
 - PROCEDURE, 5-11, 9-27
 - PUT, 2-19, 2-20, 9-29, 11-2
 - READ, 2-19, 2-23, 9-33, 11-3
 - REPLACE, 2-5, 12-2
 - RETURN, 9-34
 - REVERT, 9-34
 - REWRITE, 2-19, 2-23, 9-35
 - SIGNAL, 9-35
 - STOP, 2-20, 9-36
 - THEN, 9-22
 - WRITE, 2-19, 2-23, 9-36, 11-3
- STATIC attribute, 5-15
- STATIC storage class, 2-14, 4-2
- STATISTICS compiler option,
 14-7, G-3, G-7
- STOP statement, 2-20, 9-36

- Storage classes,
 - All types defined, A-22
 - AUTOMATIC, 2-14, 4-1
 - BASED, 4-3
 - DEFINED, 4-5
 - In general, 2-14, 4-1
 - PARAMETER, 4-6
 - STATIC, 2-14, 4-2
 - Valid, listed, 5-7
 - Storage formats, C-1
 - Storage order for arrays, 3-18
 - Storage sharing, 4-8
 - STORE_OWNER_FIELD compiler
 - option, G-3, G-8
 - STREAM attribute, 2-19, 5-15
 - Stream file, 2-17
 - Stream files, 11-3
 - Stream I/O, 2-20
 - STRING built-in function, 10-12
 - String overlays, 4-8
 - STRING pseudo-variable, 9-3
 - Structure data, 3-19
 - Structures,
 - Arrays of, 3-21
 - Arrays within, 3-21
 - As parameters, 4-7, 5-11
 - AS wholes, 3-20
 - Assignment to, 3-20, 9-2
 - Declaring, 3-20, 5-4
 - Efficiency of, 12-1
 - I/O of, 3-20, 9-19, 9-30
 - In general, 3-19
 - Level numbers in, 3-19, 5-4
 - Major, 3-19
 - Members of, 3-19
 - Overlapping, 6-2
 - Referencing, 3-20, 6-2, 6-5, 12-6
 - Shared storage and, 4-8
 - Substructures of, 3-19
 - Style of programming, 12-3
 - Subscripts,
 - Array, 3-19
 - Expressions as, 6-1
 - In structure references, 6-2, 12-6
 - Label, 3-14
 - Of based data, 4-4
 - Of labels, 5-2
 - Subscripted references, 6-1, 6-2, 6-5
 - SUBSTR built-in function, 10-13
 - SUBSTR pseudo-variable, 9-4
 - Substructures, 3-19
 - Subtraction, 7-4
 - Suppress warning messages, 14-7
 - SYMBOL editor command, 1-7
- T
- TAB format, 9-17
 - Tab stops, 11-5
 - Tables,
 - Arithmetic conversion
 - precision, 8-3
 - Characters as bit-strings, 3-11
 - Compiler options, 14-4, 14-11
 - Of picture data conversions, 8-15
 - TAN built-in function, 10-13
 - TAND built-in function, 10-13
 - TANH built-in function, 10-13
 - Text insertion, 2-4
 - Text replacement, 2-4
 - THEN statement, 9-22

TIME built-in function, 10-13
 TIME compiler option, G-3, G-7
 TITLE option, 2-17, 2-18, 9-26,
 11-3
 Tokens, 2-1
 TRANSLATE built-in function,
 10-13
 TRUNC built-in function, 10-14
 TTY Input/Output, 11-2
 Type conversion,
 In general, 2-14
 Type-conversion,
 And prefix operators, 7-4
 Arith to arith, 8-2
 Arith to bit-string, 8-4
 Arith to char-string, 8-5
 Automatic, 8-1
 Bit-string to arith, 8-6
 Bit-string to char-string, 8-7
 Causes of, 8-1
 Char-string to arith, 8-7
 Char-string to bit-string, 8-8
 During I/O, 8-8
 For arithmetic data, 7-3
 Format-controlled, 8-8
 Implicit, 8-1
 In expressions, 7-3
 In general, 8-1
 Of arguments, 9-6
 On assignment, 9-2
 On output, 9-31, 9-32
 Picture to arith, 8-13
 Picture to bit-string, 8-13
 Picture to char-string, 8-14
 Relational, 7-7
 To picture data, 8-14
 With A format, 8-11
 With B format, 8-12
 With E format, 8-10
 With F format, 8-9
 With P format, 8-13
 Type-conversions,
 Arith-to-char, 11-5
 Bit-string, 7-8

U

Undefined data, 6-5
 UNSPEC built-in function, 10-14
 UNSPEC pseudo-variable, 9-4
 UPCASE compiler option, 14-5,
 G-2, G-5
 UPDATE attribute, 2-19, 2-24,
 5-15
 Upper-to-lower-case, 14-5

V

VALID built-in function, 10-14
 VARIABLE attribute, 3-15, 5-10,
 5-16
 Variable references, 6-4
 Variables,
 Automatic, 2-14
 Based, 12-1
 Bit, 3-9
 Character, 3-7
 Declaring, 5-3
 ENTRY, 3-14
 File, 2-17, 3-17
 Fixed-point, 3-2
 Floating-point, 3-3
 In general, 2-13
 Label, 3-12
 Picture, 3-4
 Pointer, 3-10
 Scalar, defined, 3-1
 Static, 2-14
 Undefined, 12-3
 VARYING attribute, 3-8, 5-16
 Varying-length lines, 11-3
 VERIFY built-in function, 10-15

UPD4031-13A

W

WRITE statement, 2-19, 2-23,
9-36, 11-3

X

X format, 9-17

XREF compiler option, 14-6, G-3,
G-8