

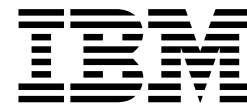
IBM Asset Transformation Workbench
v1.1



Architecture Reference Manual



IBM Asset Transformation Workbench
v1.1



Architecture Reference Manual

Note:

Before using this information and the product it supports, read the information in “Notices.”

First Edition (February 2005)

This edition applies to IBM Asset Transformation Workbench (product number 5724-L54) and to all subsequent releases and modifications until otherwise indicated in new editions.

For the latest information about this product, please refer to the Release Notes.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of IBM. Information in this document is subject to change without notice and is not guaranteed to be error-free.

You can order publications through your IBM representative or the IBM branch office serving your locality. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Licensed Materials - Property of IBM.

Product Reference: IBM Asset Transformation Workbench v1.1

Document Reference: REL7.3.07.DOC09.A

© 2005 Copyright International Business Machines Corporation. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© 2004, 2005 Relativity Technologies, Inc. All rights reserved.

RescueWare is a registered trademark of Relativity Technologies, Inc. All other brands mentioned in this document are trademarks or registered trademarks of their respective holders.

Contents

Preface

<i>Audience</i>	xi
<i>Organization</i>	xii
<i>Conventions</i>	xii
<i>Related Manuals</i>	xiii
<i>Online Help</i>	xiii

1 Supported COBOL Statements

<i>Key to tables</i>	1-1
<i>Supported VS COBOL II statements</i>	1-3
<i>COBOL language structure</i>	1-3
<i>Literals</i>	1-5
<i>Referencing names</i>	1-5
<i>COBOL program structure</i>	1-7
<i>Identification Division</i>	1-7
<i>Environment Division</i>	1-8
<i>Data Division</i>	1-10
<i>Procedure Division</i>	1-14
<i>Compile-directing statements and directives</i>	1-26

- Supported COBOL for OS/390 statements. 1-28
 - COBOL language structure 1-28
 - Literals 1-30
 - Referencing names 1-31
 - COBOL program structure 1-32
 - Identification Division 1-32
 - Environment Division 1-34
 - Data Division 1-37
 - Procedure Division 1-41
 - Intrinsic functions 1-56
 - Compile-directing statements and directives 1-58
- Supported COBOL/400 statements 1-60
 - COBOL language structure 1-60
 - Literals 1-61
 - Referencing names 1-61
 - COBOL program structure 1-62
 - Identification Division 1-63
 - Environment Division 1-64
 - Data Division 1-67
 - Procedure Division 1-71
 - Compile-directing statements and directives 1-87
- Supported Unisys ASCII COBOL statements. 1-89
 - COBOL language structure 1-89
 - Literals 1-90
 - Referencing names 1-90
 - Control Division 1-91
 - Identification Division 1-92
 - Environment Division 1-92
 - Data Division 1-96
 - Procedure Division 1-103
- Supported Unisys UCS COBOL statements. 1-117
 - COBOL language structure 1-117
 - Literals 1-118
 - Referencing names 1-118
 - Control Division 1-119
 - Identification Division 1-120
 - Environment Division 1-120

<i>Data Division</i>	1-124
<i>Procedure Division</i>	1-131
<i>Supported Unisys CDML COBOL statements</i>	1-145
<i>Data Division</i>	1-145
<i>Procedure Division</i>	1-147
<i>Update Commands</i>	1-150
<i>Support Commands</i>	1-151
<i>Supported MicroFocus COBOL statements</i>	1-153
<i>COBOL language structure</i>	1-153
<i>Literals</i>	1-154
<i>Referencing names</i>	1-154
<i>Identification Division</i>	1-155
<i>Environment Division</i>	1-156
<i>Data Division</i>	1-159
<i>Procedure Division</i>	1-165
<i>Intrinsic functions</i>	1-184
<i>Supported UTS 4000 COBOL statements</i>	1-187
<i>COBOL language structure</i>	1-187
<i>Literals</i>	1-188
<i>COBOL program structure</i>	1-189
<i>Identification Division</i>	1-189
<i>Environment Division</i>	1-190
<i>Data Division</i>	1-191
<i>Procedure Division</i>	1-194
<i>Supported HP COBOL II/XL statements</i>	1-204
<i>COBOL language structure</i>	1-204
<i>Literals</i>	1-205
<i>Referencing names</i>	1-205
<i>COBOL program structure</i>	1-206
<i>Identification Division</i>	1-207
<i>Environment Division</i>	1-208
<i>Data Division</i>	1-209
<i>Procedure Division</i>	1-212
<i>Intrinsic functions</i>	1-225
<i>Preprocessor commands</i>	1-226
<i>Supported Fujitsu COBOL85 (M Series) statements</i>	1-228
<i>COBOL language structure</i>	1-228

<i>Literals</i>	1-229
<i>Referencing names</i>	1-230
<i>COBOL program structure</i>	1-231
<i>Identification Division</i>	1-232
<i>Environment Division</i>	1-232
<i>Data Division</i>	1-236
<i>Procedure Division</i>	1-241
<i>Intrinsic functions</i>	1-255
<i>Compile-directing statements and directives</i>	1-255
<i>Supported AIM Network Database Statements</i>	1-258
<i>Supported SIEMENS COBOL statements</i>	1-260
<i>COBOL language structure</i>	1-260
<i>Literals</i>	1-262
<i>Referencing names</i>	1-263
<i>COBOL program structure</i>	1-264
<i>Identification Division</i>	1-264
<i>Environment Division</i>	1-266
<i>Data Division</i>	1-269
<i>Procedure Division</i>	1-273
<i>Intrinsic functions</i>	1-289
<i>Compile-directing statements and directives</i>	1-291

2 Supported Natural Statements

<i>Key to tables</i>	2-1
<i>Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows</i>	
<i>statements</i>	2-3
<i>Constants</i>	2-3
<i>System variables</i>	2-4
<i>Date and time system variables</i>	2-7
<i>Session parameters</i>	2-7
<i>System functions</i>	2-9
<i>Conditional expressions</i>	2-10
<i>Statements</i>	2-11
<i>SQL Statements</i>	2-28

3 Supported PL/I Statements

<i>Key to tables</i>	3-1
<i>Supported IBM Visual Age PL/I 2.0 statements</i>	3-3
<i>Problem data types</i>	3-3
<i>Program-control data types</i>	3-3
<i>Aggregate types and attributes</i>	3-4
<i>Expressions and references</i>	3-5
<i>Statements</i>	3-8
<i>Built-in functions and pseudovariables</i>	3-10

4 Supported SQL, CICS, and IMS Statements

<i>Supported SQL Statements</i>	4-2
<i>Comparisons</i>	4-2
<i>Supported CICS command statements</i>	4-9
<i>Supported IMS/Exec DLI statements</i>	4-12
<i>Supported RDMS 2200 SQL Statements</i>	4-12

5 Supported CA-IDMS DML Statements

<i>Supported CA-IDMS DML statements</i>	5-1
---	-----

Bibliography

Notices

Preface

The IBM Asset Transformation Workbench (ATW) is a suite of PC-based software products for analyzing, re-architecting, and transforming legacy applications. The products are deployed in an integrated environment with access to a common repository of program objects. Repository models serve as the basis for a rich set of diagrams, reports, and other documentation.

The ATW suite consists of customizable modules that together address the needs of organizations at every stage of legacy application evolution — maintenance/enhancement, renovation, and modernization.

Audience

This guide assumes that you are a corporate Information Technology (IT) professional with a working knowledge of the legacy platforms you are using the product to analyze. If you are transforming a legacy application, you should also have a working knowledge of the target platform.

Organization

This guide contains the following chapters:

- Chapter 1, “Supported COBOL Statements,” describes the COBOL statements supported by the ATW Application Architect product.
- Chapter 2, “Supported Natural Statements,” describes the Natural statements supported by the ATW Application Architect product.
- Chapter 3, “Supported PL/I Statements,” describes the PL/I statements supported by the ATW Application Architect product.
- Chapter 4, “Supported SQL, CICS, and IMS Statements,” describes SQL, CICS, and IMS statements supported by the ATW Application Architect product.
- Chapter 5, “Supported CA-IDMS DML Statements,” describes IDMS Data Manipulation Language statements supported by the ATW Application Architect product.

Conventions

This guide uses the following typographic conventions:

- **Bold type** — Indicates a specific area within the graphical user interface, such as a button on a screen, a window name, or a command or function.
- *Italic type* — Indicates a new term. Also indicates a document title. Occasionally, italic type is used for emphasis.
- `Monospace type` — Indicates computer programming code.
- **Bold monospace type** — Indicates input you type on the computer keyboard.
- **1A/1B, 2A/2B** — In task descriptions, indicates mutually exclusive steps: perform step A or step B, but not both.

Related Manuals

This document is part of a complete set of ATW manuals. Together they provide all the information you need to get the most out of the system.

- *Getting Started* introduces ATW. This guide provides an overview of the workbench tools and discusses basic concepts. It describes how to install the product and how to manage licenses. It also describes how to use common product features.
- *Preparing Projects* describes how to set up ATW projects. This guide describes how to load applications in the repository and how to use reports and other tools to ensure that the entire application is available for analysis.
- *Analyzing Projects* describes how to analyze applications at the project level. This guide describes how to create diagrams of applications and how to perform impact analysis across applications. It also describes how to estimate project complexity and effort, and how to create a project dictionary.
- *Analyzing Programs* describes how to analyze applications at the program level. This guide describes how to use HyperView tools to view programs interactively and perform program analysis in stages. It also describes how to analyze procedure and data flows, search the repository, and extract business rules with HyperView.
- *Profiling Projects* describes how to create and browse Web-generated views of the repositories in your organization.
- *Creating Components* describes how to extract program components from a legacy application.
- *Parser Reference Manual* describes legacy constructions supported by Application Analyzer in reference format.

Online Help

In addition to the manuals provided with the system, you can learn about the product using the integrated online help. All GUI-based tools include a standard Windows **Help** menu.

You can display:

- The entire help system, with table of contents, index, and search tool, by selecting **Help: Help Topics**.
- Help about a particular ATW window by clicking the window and pressing the **F1** key.

Many ATW tools have *guides* that you can use to get started quickly in the tool. The guides are help-like systems with hyperlinks that you can use to access functions otherwise available only in menus and other program controls.

To open the guide for a tool, choose **Guide** from the **View** menu. Use the table of contents in the **Page** drop-down to navigate quickly to a topic.

Supported COBOL Statements



This chapter describes the Common Business-Oriented Language (COBOL) statements supported by ATW for component extraction. For a list of supported legacy versions, refer to the *Release Notes*.

Key to tables

The tables below contain detailed information on ATW component extraction support. The first column of each table presents COBOL keywords. Next, if needed, in the ‘Format’ column, the possible forms of usage for each keyword are explained. These forms are represented in the standard grammar notation, where

- the optional arguments are written in square brackets ‘[arguments]’;
- the arguments which can not be omitted are listed in angle brackets <arguments>, and
- the vertical line ‘|’ means ‘or’ logical connective.

In this way, the notation <arg1 | arg2 | arg3> means that one of the arg1, arg2 or arg3 is required. Other columns provide the following information:

- **Pars.** — ATW Parser. The possible values in this column are:
 - (V)erified — parsed and prepared for further processing. Possibly is supported by other ATW tools. Does not initiate any errors or warnings during the verification phase.
 - (V)erification (O)nly — parsed and skipped immediately. Further processing is not possible, i.e. not supported by ATW tools. Verification warning is initiated.
 - (N)ot Verified — if the ATW parser finds such an entity it stops and verification is considered as erroneous.

Note: Object-oriented COBOL statements are not supported.

- **CE** — Component Extraction. CE (S)upport means that the entity can be correctly sliced. If the entity is (N)ot supported and is represented in the slicing code, the obtained program may not work correctly. The ‘Gen.’ column gives information on ATW support for:
 - Structure-Based Component Extraction
 - Computation-Based Component Extraction
 - Entry Point Isolation
 - Dead Code Elimination

The ‘DB’ is referenced to Domain-Based Component Extraction support. The tables below also provide information on ‘Read’ and ‘Write’ variables. This is the knowledge which is used by ATW to simplify the slicing program. If the variable is situated in ‘Read’ column, then ATW ‘knows’ that it is used when corresponding statement is executed. If it is in the ‘Write’ column, it will be modified when operation is performed.

Supported VS COBOL II statements

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S
NULL / NULLS	V	S

Special registers

Registers	Parser	CE
ADDRESS OF	V	PS
DEBUG ITEM	V	N
LENGTH OF	V	S
LINAGE-COUNTER	V	N
RETURN-CODE	V	N
SHIFT OUT / SHIFT IN	V	N
SORT-CONTROL	V	N

1-4 Supported COBOL Statements
Supported VS COBOL II statements

Registers	Parser	CE
SORT-CORE-SIZE	V	N
SORT-FILE-SIZE	V	N
SORT-MESSAGE	V	N
SORT-MODE-SIZE	V	N
SORT-RETURN	V	N
TALLY	V	S
WHEN COMPILED	V	N

Literals

Literal type	Format	Parser	CE
Nonnumeric literal	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
	<ul style="list-style-type: none"> Format 2: with apostrophes Example: 'THIS ISN'T WRONG' 	V	S
	<ul style="list-style-type: none"> Format 3: with double-byte characters "EBCDIC-data<D1D2>EBCDIC-data" 	N	N
	<ul style="list-style-type: none"> Format 4: hexadecimal notation X"hexadecimal-digits" 	V	S
Numeric literals	<ul style="list-style-type: none"> Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
DBCS literals	<ul style="list-style-type: none"> Format G"<D1D2D3>" 	V	S
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format text-name-1 [<IN OF> library-name-1] 	V	S

1-6 Supported COBOL Statements
Supported VS COBOL II statements

Division	Format	Parser	CE
to Procedure Division	<ul style="list-style-type: none"> Format 1 <i>paragraph-name-1</i> [<IN OF> <i>section-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2 <i>section-name-1</i> 	V	S
to Data Division	<ul style="list-style-type: none"> Format 1: simple data reference <i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> Format 2: subscripting <i>data-name-1</i> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) (<i>leftmost-character-position: [length]</i>) 	V	S
	<ul style="list-style-type: none"> Format 3 <<i>condition-name-1</i> <i>data-name-1</i>> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 4 LINAGE-COUNTER [<IN OF> <i>file-name-2</i>] 	V	S
Condition names	<ul style="list-style-type: none"> Format 1: Data Division <i>condition-name-1</i> [<IN OF> <i>data-name-1</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) 	V	S
	<ul style="list-style-type: none"> Format 2: Special-Names paragraph <i>condition-name-1</i> [<IN OF> <i>mnemonic-name-1</i>] 	V	S
Subscript	<ul style="list-style-type: none"> Format <<i>integer-1</i> ALL <i>data-name-3</i> [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>]> 	V	S
Reference modification	<ul style="list-style-type: none"> Format <<i>data-name-1</i> FUNCTION <i>function-name-1</i> (<i>arguments</i>)> (<i>leftmost-character-position: [length]</i>) 	V	S
	FUNCTION <i>function-name-1</i> (<i>arguments</i>)>	VO	N
	(<i>leftmost-character-position: [length]</i>)	V	S

COBOL program structure

Program type	Format	Parser	CE
Single program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> END-PROGRAM <i>program-name-1</i> .	V	S
Nested program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> <i>nested source program</i> END-PROGRAM <i>program-name-1</i> .	N	N

Identification Division

Paragraphs	Format	Parser	CE
Program Identification Division	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name</i> [[IS] <RECURSIVE COMMON [INITIAL] INITIAL [COMMON]> [PROGRAM]]. [AUTHOR. <i>comment-entry</i>] [INSTALLATION. <i>comment-entry</i>] [DATE-WRITTEN. <i>comment-entry</i>] [DATE-COMPILED. <i>comment-entry</i>] [SECURITY. <i>comment-entry</i>]	V	S

1-8 Supported COBOL Statements
Supported VS COBOL II statements

Environment Division

Input-output section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: QSAM/SAM/VSAM sequential file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [[ORGANIZATION [IS]] SEQUENTIAL] [PADDING [CHARACTER] [IS] < <i>data-name-5</i> <i>literal-2</i> >] [RECORD DELIMITER [IS] <STANDARD-1 <i>assignment-name-2</i> >] [ACCESS [MODE] [IS] SEQUENTIAL] [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 2: VSAM indexed file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> <i>literal-1</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] INDEXED [ACCESS [MODE] [IS] <SEQUENTIAL RANDOM DYNAMIC>] RECORD [KEY] [IS] <i>data-name-2</i> [PASSWORD [IS] <i>data-name-6</i>] ALTERNATE RECORD [KEY] [IS] <i>data-name-3</i> [PASSWORD [IS] <i>data-name-7</i>] [[WITH] DUPLICATES] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 3: VSAM relative file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> <i>literal-1</i> >	V	S
[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] RELATIVE [ACCESS [MODE] [IS] <SEQUENTIAL [RELATIVE [KEY] [IS] <i>data-name-4</i>] <RANDOM DYNAMIC> RELATIVE [KEY] [IS] <i>data-name-4</i> >] [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N	

Paragraphs and entries	Format	Parser	CE
I-O-CONTROL paragraph	<ul style="list-style-type: none"> • Format 1: QSAM/SAM I-O I-O-CONTROL. <RERUN ON <assignment-name-1 file-name-1> [EVERY] <integer-1 RECORDS END [OF] <REEL UNIT>> [OF] file-name-1 SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4 MULTIPLE FILE [TAPE] [CONTAINS] file-name-5 [POSITION] integer-2 APPLY WRITE-ONLY [ON] file-name-2>. 	VO	N
	<ul style="list-style-type: none"> • Format 2: VSAM I-O I-O-CONTROL. <RERUN ON <assignment-name-1 file-name-1> [EVERY] integer-1 RECORDS [OF] file-name-1 SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4> 	VO	N
	<ul style="list-style-type: none"> • Format 3: sort-merge I-O I-O-CONTROL. [RERUN [ON] assignment-name-1] SAME <RECORD SORT SORT-MERGE> [AREA] [FOR] file-name-3 file-name-4. 	VO	N

1-10 Supported COBOL Statements
Supported VS COBOL II statements

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COM-PUTER paragraph	<ul style="list-style-type: none"> Format OBJECT-COMPUTER. [<i>computer-name</i> [MEMORY [SIZE] <i>integer</i> <WORDS CHARACTERS MODULES>] [[PROGRAM] [COLLATING] SEQUENCE [IS] <i>alphabet-name</i>] [SEGMENT-LIMIT [IS] <i>priority-number</i>].] 	VO	N
SOURCE-COM-PUTER paragraph	<ul style="list-style-type: none"> Format SOURCE-COMPUTER. [<i>computer-name</i> [[WITH] DEBUGGING MODE].] 	VO	N
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> Format SPECIAL-NAMES. [<i>environment-name-1</i> [IS] <i>mnemonic-name-1</i> <i>environment-name-2</i> <[IS] <i>mnemonic-name-2</i> <i>entry-1</i> <i>entry-1</i>>] where <i>entry-1</i> is: <i>entry-1</i>: <ON [STATUS] [IS] <i>condition-1</i> [OFF [STATUS] [IS] <i>conditional-2</i>] OFF [STATUS] [IS] <i>condition-2</i> [ON [STATUS] [IS] <i>conditional-1</i>]> 	V	PS
	[ALPHABET <i>alphabet-name-1</i> [IS] <STANDARD-1 STANDARD-2 NATIVE EBCDIC <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i>]>]	VO	N
	[SYMBOLIC [CHARACTERS] <i>symbolic-character</i> [ARE IS] <i>integer-1</i> [IN] <i>alphabet-name-2</i>]	VO	N
	[CLASS <i>class-name-1</i> [IS] <i>literal-4</i> [<THROUGH THRU> <i>literal-5</i>]	VO	N
	[CURRENCY [SIGN] [IS] <i>literal-6</i> [[WITH] PICTURE SYMBOL <i>literal-7</i>]	VO	N
	[DECIMAL-POINT [IS] COMMA] [.]	V	S

Data Division

Entries and clauses	Format	Parser	CE
Data Division	<ul style="list-style-type: none"> Format: program and method DATA DIVISION DATA DIVISION. [FILE SECTION. [<i>file-description-entry</i> <i>record-description-entry</i>]] [WORKING-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [LINKAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] 	V	S

Entries and clauses	Format	Parser	CE
File Description (FD) Entry	<ul style="list-style-type: none"> Format 1: sequential files FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i>] [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED <i>data-name-2</i>>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [RECORDING [MODE] [IS] <i>mode</i>] [CODE-SET [IS] <i>alphabet-name</i>]. 	V	S
	<ul style="list-style-type: none"> Format 2: relative and indexed files FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i>] [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED <i>data-name-2</i>>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>]. 	V	S
	<ul style="list-style-type: none"> Format 3: sort/merge files SD <i>file-name-1</i> [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i>] [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED <i>data-name-2</i>>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [CODE-SET [IS] <i>alphabet-name</i>]. 	V	S

1-12 Supported COBOL Statements
Supported VS COBOL II statements

Entries and clauses	Format	Parser	CE
Data Description Entry	<ul style="list-style-type: none"> Format 1 <i>level-number <data-name-1 FILLER></i> <i>[redefines-clause] [blank-when-zero-clause] [external-clause] [global-clause] [justified-clause] [occurs-clause] [picture-clause] [sign-clause] [synchronized-clause] [usage-clause] [value-clause]</i> 	V	S
	<ul style="list-style-type: none"> Format 2 <i>66 data-name-1 renames-clause.</i> 	V	S
	<ul style="list-style-type: none"> Format 3 <i>88 condition-name value-clause.</i> 	V	S
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> Format <i>BLANK [WHEN] <ZERO ZEROS ZEROES></i> 	V	S
DATE FORMAT clause	<ul style="list-style-type: none"> Format <i>DATE FORMAT [IS] date-pattern</i> 	VO	N
EXTERNAL clause		V	N
GLOBAL clause		V	N
JUSTIFIED clause	<ul style="list-style-type: none"> Format <i><JUSTIFIED JUST> [RIGHT]</i> 	VO	N
OCCURS clause	<ul style="list-style-type: none"> Format 1: fixed-length tables <i>OCCURS integer-2 [TIMES] [<ASCENDING DESCENDING> [KEY] [IS] data-name-2] [INDEXED [BY] index-name-1]</i> 	V	S
	<ul style="list-style-type: none"> Format 2: variable-length tables <i>OCCURS integer-1 TO integer-2 [TIMES] DEPENDING [ON] data-name-1 [<ASCENDING DESCENDING> [KEY] [IS] data-name-2] [INDEXED [BY] index-name-1]</i> 	V	S
PICTURE clause	<ul style="list-style-type: none"> Format <i><PICTURE PIC> [IS] character-string</i> 	V	S
REDEFINES clause	<ul style="list-style-type: none"> Format <i>level-number <data-name-1 FILLER> REDEFINES data-name-2</i> 	V	S

Entries and clauses	Format	Parser	CE
RENA-MES clause	<ul style="list-style-type: none"> Format 66 <i>data-name-1</i> RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>] 	V	S
SIGN clause	<ul style="list-style-type: none"> Format SIGN [IS] <LEADING TRAILING>[SEPARATE CHARACTER] 	V	N
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZED SYNC> [LEFT RIGHT] 	VO	N
USAGE clause	<ul style="list-style-type: none"> Format USAGE [IS] <BINARY COMP COMP-1 COMP-2 COMP-3 COMP-4 COMPUTATIONAL COMPUTATIONAL-1 COMPUTATIONAL-2 COMPUTATIONAL-3 COMPUTATIONAL-4 DISPLAY DISPLAY-1 INDEX PACKED-DECIMAL POINTER > 	V	S
VALUE clause	<ul style="list-style-type: none"> Format 1: literal value VALUE [IS] <i>literal</i> 	V	S
	<ul style="list-style-type: none"> Format 2: condition-name value 88 <i>condition-name-1</i> <VALUE [IS] VALUES [ARE]> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]. 	V	S
	<ul style="list-style-type: none"> Format 3: NULL value VALUE [IS] <NULL NULLS> 	V	S

Procedure Division

Arithmetic operators

Binary operation	Meaning	Format	Parser	CE			
				Gen.	Read variables	Written variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

Conditional Expressions

Conditions	Format	Parser	CE			
			Gen.	Read variables	Written variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1</i> [IS] [NOT] <NUMERIC ALPHABETIC ALPHABETIC-LOWER ALPHABETIC-UPPER <i>class-name</i> DBCS KANJI> 	V	S	<i>id-1</i>		S
		VO	N			N
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] <[NOT] <GREATER [THAN] LESS [THAN] EQUAL [TO] > GREATER [THAN] OR EQUAL [TO] LESS [THAN] OR EQUAL [TO]> <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>		S
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	<i>operand-1</i>		S
Switch-status condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT <i>condition-1</i> 	V	S	<i>condition-1</i>		S
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1</i> <AND OR> <i>condition-2</i> 	V	S	<i>condition-1</i> <i>condition-2</i>		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format <i>relation-condition</i> <AND OR> [NOT] [<i>relational-operator</i>] <i>object</i> 	V	S	<i>relation-condition</i> <i>object</i>		S

1-16 Supported COBOL Statements
Supported VS COBOL II statements

Statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format PROCEDURE DIVISION [USING <i>data-name-1</i>]. 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N
PROCEDURE DIVISION structure	<ul style="list-style-type: none"> Format: procedure division header [DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES.] <i>section-name</i> SECTION [<i>priority-number</i>]. [<i>paragraph-name</i>. [<i>sentence</i>]] 					
DECLARATIVES section	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FROM <<i>mnemonic-name</i> <i>environment-name</i>>] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE DAY DAY-OF-WEEK TIME> 	V	S		<i>id</i>	N
ADD	<ul style="list-style-type: none"> Format 1 ADD <<i>id-1</i> <i>literal</i>> TO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <<i>id-1</i> <i>literal-1</i>> TO <<i>id-2</i> <i>literal-1</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ALTER	<ul style="list-style-type: none"> Format ALTER <i>procedure-name-1</i> TO [PROCEED TO] <i>procedure-name-2</i> 	V	S			N
CALL	<ul style="list-style-type: none"> Format 1: with ON OVERFLOW CALL <<i>id-1</i> <i>literal-1</i>> [USING <[[BY] REFERENCE] <[ADDRESS OF] <i>id-2</i> <i>id-3</i> <i>file-name-1</i>> [BY] CONTENT <[ADDRESS OF LENGTH OF] <i>id-3</i> <i>literal-2</i>>>] [ON] OVERFLOW <i>imperative-stmt-1</i>] [END-CALL] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with ON EXCEPTION CALL <<i>id-1</i> <i>literal-1</i>> [USING <[[BY] REFERENCE] <[ADDRESS OF] <i>id-2</i> <i>id-3</i> <i>file-name-1</i>> [BY] CONTENT <[ADDRESS OF LENGTH OF] <i>id-3</i> <i>literal-2</i>>>] [[ON] EXCEPTION <i>imperative-stmt-1</i>] [NOT [ON] EXCEPTION <i>imperative-stmt-2</i>] [END-CALL] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i>	<i>id-2</i>	S
CANCEL	<ul style="list-style-type: none"> Format CANCEL <<i>id-1</i> <i>literal-1</i>> 	V	S			N
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> [<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND] [WITH] <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files CLOSE <i>file-name-1</i> [[WITH] LOCK] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <i>id-1</i> [ROUNDED] [EQUAL] <i>arithmetic-expr</i> [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-COMPUTE] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S
CONTINUE	<ul style="list-style-type: none"> Format CONTINUE 	V	S			N

1-18 Supported COBOL Statements
Supported VS COBOL II statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DELETE	<ul style="list-style-type: none"> Format DELETE <i>file-name-1</i> [RECORD] [[INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-DELETE] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISPLAY	<ul style="list-style-type: none"> Format DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <<i>mnemonic-name-1</i> <i>environment-name-1</i>>] [[WITH] NO ADVANCING] 	V	S	<i>id-1</i>		S
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <<i>id-1</i> <i>literal-1</i>> INTO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-DIVIDE] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <<i>id-1</i> <i>literal-1</i>> INTO <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-DIVIDE] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <<i>id-1</i> <i>literal-1</i>> BY <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-DIVIDE] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <<i>id-1</i> <i>literal-1</i>> INTO <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] REMAINDER <i>id-4</i> [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-DIVIDE] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i> <i>id-4</i>	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <<i>id-1</i> <i>literal-1</i>> BY <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] REMAINDER <i>id-4</i> [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-DIVIDE] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i> <i>id-4</i>	S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ENTRY	<ul style="list-style-type: none"> Format ENTRY <i>literal-1</i> [USING <i>id-1</i>] 	V	S	<i>id-1</i>		N
EVALUATE	<ul style="list-style-type: none"> Format EVALUATE <<i>id-1</i> <i>literal-1</i> <i>expr-1</i> TRUE FALSE> [ALSO <<i>id-2</i> <i>literal-2</i> <i>expr-2</i> TRUE FALSE>] WHEN <i>phrase-1</i> [ALSO <i>phrase-2</i> <i>imperative-stmt-1</i>] [WHEN OTHER <i>imperative-stmt-2</i>] [END-EVALUATE] Phrases <i>phrase-1</i> and <i>phrase-2</i> should be represented in the following form: <ANY <i>condition-1</i> TRUE FALSE [NOT] <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> [<THROUGH THRU> <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>>]> 	V	S	<i>id-1</i> <i>expr-1</i> <i>id-2</i> <i>expr-2</i> <i>condition-1</i> <i>id-3</i> <i>id-4</i> <i>arithmetic-expr1</i> <i>arithmetic-expr2</i>		N
EXIT	<ul style="list-style-type: none"> Format <i>paragraph-name</i>. EXIT. 	V	S			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
GO-BACK	<ul style="list-style-type: none"> Format GOBACK 	V	S			N
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO [TO] <i>procedure-name-1</i> 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO [TO] <i>procedure-name-1</i> DEPENDING [ON] <i>id-1</i> 	V	S	<i>id-1</i>		N
	<ul style="list-style-type: none"> Format 3: altered <i>paragraph-name</i>. GO [TO]. 	V	S			N
	<ul style="list-style-type: none"> Format 4: MORE-LABELS GO [TO] MORE-LABELS 	N	N			N

1-20 Supported COBOL Statements
Supported VS COBOL II statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
IF	<ul style="list-style-type: none"> Format IF <i>condition-1</i> THEN <<i>stmt-1</i> NEXT SENTENCE> [ELSE <<i>stmt-2</i> NEXT SENTENCE>] [END-IF] 	V	S	<i>condition-1</i>		N
INITIALIZE	<ul style="list-style-type: none"> Format INITIALIZE <i>id-1</i> [REPLACING <ALPHABETIC ALPHANUMERIC NUMERIC ALPHANUMERIC-EDITED NUMERIC-EDITED DBCS EGCS> [DATA] BY <<i>id-2</i> <i>literal-1</i>>] 	V	S	<i>id-2</i>	<i>id-1</i>	N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-5</i> <i>id-4</i> <i>id-3</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 4 INSPECT <i>id-1</i> CONVERTING <<i>id-6</i> <i>literal-4</i>> TO <<i>id-7</i> <i>literal-5</i>> [BEFORE AFTER] [INITIAL] <<i>id-4</i> <i>literal-2</i>> 	V	S	<i>id-1</i> <i>id-6</i> <i>id-7</i> <i>id-4</i>	<i>id-1</i>	N
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> 	V	S	<i>file-name-1</i> <i>data-name-1</i> <i>file-name-2</i> <i>file-name-3</i>	<i>file-name-4</i>	N
	<ul style="list-style-type: none"> [[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>] 	VO	N			N
	<ul style="list-style-type: none"> USING <i>file-name-2</i> <i>file-name-3</i> <OUTPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i>> 	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <id-1 literal-1> TO id-2 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> id-1 TO id-2 	V	S	id-1	id-2	N
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <id-1 literal-1> BY id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1 id-2	id-3	S
OPEN	<ul style="list-style-type: none"> Format 1: QSAM/SAM files OPEN <INPUT file-name-1 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
	<ul style="list-style-type: none"> [REVERSED [WITH] NO REWIND] 	VO	N			
	<ul style="list-style-type: none"> OUTPUT file-name-2 	V	S			
	<ul style="list-style-type: none"> [[WITH] NO REWIND] 	VO	N			
	<ul style="list-style-type: none"> I-O file-name-3 EXTEND file-name-4> 	V	S			
	<ul style="list-style-type: none"> Format 2: VSAM files OPEN <INPUT file-name-1 OUTPUT file-name-2 I-O file-name-3 EXTEND file-name-4> 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N

1-22 Supported COBOL Statements
Supported VS COBOL II statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM <procedure-name-1 [<THROUGH THRU> procedure-name-2] imperative-stmt-1 END-PERFORM> 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM <procedure-name-1 [<THROUGH THRU> procedure-name-2] <id-1 integer-1> TIMES imperative-stmt-1 END-PERFORM> 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	condition-1		N
	<ul style="list-style-type: none"> Format 4: with UNTIL phrase, without END-PERFORM PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] [[WITH] TEST <BEFORE AFTER>] UNTIL condition-1 	V	S	condition-1		N
	<ul style="list-style-type: none"> Format 5: with VARYING phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	id-3 id-4 condition-1	id-2	N
	<ul style="list-style-type: none"> Format 6: with VARYING phrase, without END-PERFORM PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 [AFTER <id-5 index-name-3> FROM <id-6 index-name-4 literal-3> BY <id-7 literal-4> UNTIL condition-2] <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-2 	N	N			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
READ	<ul style="list-style-type: none"> Format 1: sequential retrieval READ <i>file-name-1</i> 	V	S	<i>file-name-1</i>	<i>file-name-1 id-1</i>	N
	<ul style="list-style-type: none"> [NEXT PREVIOUS] [RECORD] 	VO	N			
	<ul style="list-style-type: none"> [INTO <i>id-1</i>] [[AT] END <i>imperative-stmt-1</i>] [NOT [AT] END <i>imperative-stmt-2</i>] [END-READ] 	V	S			
	<ul style="list-style-type: none"> Format 2: random retrieval READ <i>file-name-1</i> [RECORD] [INTO <i>id-1</i>] [KEY [IS] <i>data-name-1</i>] [INVALID [KEY] <i>imperative-stmt-3</i>] [NOT INVALID [KEY] <i>imperative-stmt-4</i>] [END-READ] 	V	S	<i>file-name-1 data-name-1</i>	<i>file-name-1 id-1</i>	N
RELEASE	<ul style="list-style-type: none"> Format RELEASE <i>record-name-1</i> [FROM <i>id-1</i>] 	V	S	<i>record-name-1 id-1</i>	<i>record-name-1</i>	N
RETURN	<ul style="list-style-type: none"> Format RETURN <i>file-name-1</i> [RECORD] [INTO <i>id-1</i>] [[AT] END <i>imperative-stmt-1</i>] [NOT [AT] END <i>imperative-stmt-2</i>] [END-RETURN] 	V	S	<i>file-name-1</i>	<i>id-1</i>	N
RE-WRITE	<ul style="list-style-type: none"> Format REWRITE <i>file-name-1</i> [FROM <i>id-1</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-REWRITE] 	V	S	<i>file-name-1 id-1</i>	<i>file-name-1</i>	N
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [[AT] END <i>imperative-stmt-1</i>] WHEN <i>condition-1</i> <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1 condition-1</i>	<i>id-2</i>	N

1-24 Supported COBOL Statements
Supported VS COBOL II statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SEARCH	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [[AT] END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> [IS] EQUAL [TO] <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> [IS] EQUAL [TO] <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>data-name-1</i>		N
SET	<ul style="list-style-type: none"> Format 1: basic table handling SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-2</i> integer-1> 	V	S	<i>index-name-2</i> <i>id-2</i>	<i>index-name-1</i> <i>id-1</i>	S
	<ul style="list-style-type: none"> Format 2: adjusting indexes SET <i>index-name-3</i> <UP BY DOWN BY> <<i>id-3</i> integer-2> 	V	S	<i>index-name-3</i> <i>id-3</i>	<i>index-name-3</i>	N
	<ul style="list-style-type: none"> Format 3: external switches SET <i>mnemonic-name-1</i> TO <ON OFF> 	V	S		<i>mnemonic-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: condition-names SET <i>condition-name-1</i> TO TRUE 	V	S		<i>condition-variable</i>	S
	<ul style="list-style-type: none"> Format 5: USAGE IS POINTER data items SET <<i>id-4</i> ADDRESS OF <i>id-5</i>> TO <<i>id-6</i> ADDRESS OF <i>id-7</i> NULL NULLS> 	V	S	<i>id-6</i> <i>id-7</i>	<i>id-4</i> <i>id-5</i>	N
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> [[WITH] DUPLICATES [IN] [ORDER]] 	V	S			N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N			N
	<USING <i>file-name-2</i> INPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i> > <GIVING <i>file-name-3</i> OUTPUT PROCEDURE [IS] <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i> >	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [KEY [IS] <EQUAL [TO] = LESS [THAN] < GREATER [THAN] > NOT LESS [THAN] NOT < NOT GREATER [THAN] NOT > LESS [THAN] OR EQUAL [TO] <= GREATER [THAN] OR EQUAL [TO] >= > <i>data-name-1</i>] [[INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-START] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N
STRING	<ul style="list-style-type: none"> Format STRING <<i>id-1</i> <i>literal-1</i>> DELIMITED [BY] <<i>id-2</i> <i>literal-2</i> SIZE> INTO <i>id-3</i> [[WITH] POINTER <i>id-4</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-STRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>	<i>id-3</i> <i>id-4</i>	S
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <<i>id-1</i> <i>literal-1</i>> FROM <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <<i>id-1</i> <i>literal-1</i>> FROM <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> <i>id-1</i> FROM <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N
UN-STRING	<ul style="list-style-type: none"> Format UNSTRING <i>id-1</i> [DELIMITED [BY] [ALL] <<i>id-2</i> <i>literal-1</i>> [OR [ALL] <<i>id-3</i> <i>literal-2</i>>]] INTO <i>id-4</i> [DELIMITER [IN] <i>id-5</i>] [COUNT [IN] <i>id-6</i>] [[WITH] POINTER <i>id-7</i>] [TALLYING [IN] <i>id-8</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-UNSTRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-7</i> <i>id-8</i>	<i>id-4</i> <i>id-5</i> <i>id-6</i> <i>id-7</i> <i>id-8</i>	S

1-26 Supported COBOL Statements
Supported VS COBOL II statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
WRITE	<ul style="list-style-type: none"> Format 1: VSAM sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: QSAM/SAM sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [<<BEFORE AFTER> [ADVANCING] <<<i>id-1</i> <i>integer-1</i>> [LINE LINES] <i>mnemonic-name-1</i> PAGE>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	VO	N			N
	<ul style="list-style-type: none"> Format 3: VSAM indexed and relative files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N

Compile-directing statements and directives

Statements and directives	Format	Parser	CE
BASIS	<ul style="list-style-type: none"> Format [<i>sequence-number</i>] BASIS <<i>basis-name</i> <i>literal-1</i>> 	VO	N
CBL (PROCESS)	<ul style="list-style-type: none"> Format <CBL PROCESS> [<i>options-list</i>] 	VO	N
*CONTROL (*CBL)	<ul style="list-style-type: none"> Format < *CONTROL *CBL> <SOURCE NOSOURCE LIST NOLIST MAP NOMAP> 	VO	N

Statements and directives	Format	Parser	CE
COPY	<ul style="list-style-type: none"> Format COPY <text-name-1 literal-1> [<OF IN> <library-name literal-2>] [SUPPRESS] [REPLACING operand-1 BY operand-2] 	V	S
DELETE	<ul style="list-style-type: none"> Format [sequence-number] DELETE sequence-number-field 	VO	N
EJECT	<ul style="list-style-type: none"> Format EJECT [.] 	VO	N
ENTER	<ul style="list-style-type: none"> Format ENTER language-name-1 [routine-name-1]. 	VO	N
INSERT	<ul style="list-style-type: none"> Format [sequence-number] INSERT sequence-number-field 	VO	N
READY or RESET TRACE	<ul style="list-style-type: none"> Format <READY RESET> TRACE. 	VO	N
REPLACE	<ul style="list-style-type: none"> Format 1 REPLACE ==pseudo-text-1== BY ==pseudo-text-2==. 	V	S
	<ul style="list-style-type: none"> Format 2 REPLACE OFF. 	V	S
SERVICE LABEL	<ul style="list-style-type: none"> Format SERVICE LABEL 	VO	N
SERVICE RELOAD	<ul style="list-style-type: none"> Format SERVICE RELOAD id-1 	VO	N
SKIP1/2/3	<ul style="list-style-type: none"> Format <SKIP1 SKIP2 SKIP3>. 	VO	N
TITLE	<ul style="list-style-type: none"> Format TITLE <i>literal</i>. 	VO	N

1-28 Supported COBOL Statements
Supported COBOL for OS/390 statements

Statements and directives	Format	Parser	CE
USE	<ul style="list-style-type: none"> Format 1: EXCEPTION ERROR declarative USE [GLOBAL] AFTER [STANDARD] <EXCEPTION ERROR> PROCEDURE [ON] <file-name-1 INPUT OUTPUT I-O EXTEND> 	VO	N
	<ul style="list-style-type: none"> Format 2: LABEL declarative USE [GLOBAL] AFTER [STANDARD] [BEGINNING ENDING] [FILE REEL UNIT] LABEL PROCEDURE [ON] <file-name-1 INPUT OUTPUT I-O EXTEND> 	VO	N
	<ul style="list-style-type: none"> Format 3: DEBUGGING declarative USE [FOR] DEBUGGING [ON] <procedure-name-1 ALL PROCEDURES> • 		

Supported COBOL for OS/390 statements

Note: Object-oriented COBOL statements are not supported.

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S
NULL / NULLS	V	S

Special registers

Registers	Parser	CE
ADDRESS OF	V	PS
DEBUG ITEM	V	N
LENGTH OF	V	S
LINAGE-COUNTER	V	N
RETURN-CODE	V	N
SHIFT OUT / SHIFT IN	V	N
SORT-CONTROL	V	N
SORT-CORE-SIZE	V	N
SORT-FILE-SIZE	V	N
SORT-MESSAGE	V	N
SORT-MODE-SIZE	V	N
SORT-RETURN	V	N
TALLY	V	S
WHEN COMPILED	V	N

1-30 Supported COBOL Statements
Supported COBOL for OS/390 statements

Literals

Literal type	Format	Parser	CE
Nonnumeric literal	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
	<ul style="list-style-type: none"> Format 2: with apostrophes Example: 'THIS ISN'T WRONG' 	V	S
	<ul style="list-style-type: none"> Format 3: with double-byte characters "EBCDIC-data<D1D2>EBCDIC-data" 	N	N
	<ul style="list-style-type: none"> Format 4: hexadecimal notation X"hexadecimal-digits" 	V	S
	<ul style="list-style-type: none"> Format 5: null-terminated Z"dddd" 	VO	N
Numeric literals	<ul style="list-style-type: none"> Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
DBCS literals	<ul style="list-style-type: none"> Format 1: G"<D1D2D3>" 	V	S
	<ul style="list-style-type: none"> Format 2: N"<D1D2D3>" 	N	N
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format <i>text-name-1</i> [<IN OF> <i>library-name-1</i>] 	V	S
to Procedure Division	<ul style="list-style-type: none"> Format 1: <i>paragraph-name-1</i> [<IN OF> <i>section-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2: <i>section-name-1</i> 	V	S
to Data Division	<ul style="list-style-type: none"> Format 1: simple data reference <i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> Format 2: subscripting <i>data-name-1</i> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) (<i>leftmost-character-position: [length]</i>) 	V	S
	<ul style="list-style-type: none"> Format 3: <<i>condition-name-1</i> <i>data-name-1</i>> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 4: LINAGE-COUNTER [<IN OF> <i>file-name-2</i>] 	V	S
Condition names	<ul style="list-style-type: none"> Format 1: Data Division <i>condition-name-1</i> [<IN OF> <i>data-name-1</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) 	V	S
	<ul style="list-style-type: none"> Format 2: Special-Names paragraph <i>condition-name-1</i> [<IN OF> <i>mnemonic-name-1</i>] 	V	S
Subscript	<ul style="list-style-type: none"> Format <<i>integer-1</i> ALL <i>data-name-3</i> [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>]> 	V	S
Reference modification	<ul style="list-style-type: none"> Format <<i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> FUNCTION <i>function-name-1</i> (<i>arguments</i>)> 	VO	N
	<ul style="list-style-type: none"> (<i>leftmost-character-position: [length]</i>) 	V	S

1-32 Supported COBOL Statements
Supported COBOL for OS/390 statements

COBOL program structure

Program type	Format	Parser	CE
Single program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> END-PROGRAM <i>program-name-1</i> .	V	S
Nested program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> <i>nested source program</i> END-PROGRAM <i>program-name-1</i> .	N	N

Identification Division

Paragraphs	Format	Parser	CE
Program Identification Division	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name</i> [[IS] <RECURSIVE COMMON [INITIAL] INITIAL [COMMON]> [PROGRAM]]. [AUTHOR. <i>[comment-entry]</i>] [INSTALLATION. <i>[comment-entry]</i>] [DATE-WRITTEN. <i>[comment-entry]</i>] [DATE-COMPILED. <i>[comment-entry]</i>] [SECURITY. <i>[comment-entry]</i>]	V	S
Class Identification Division	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. CLASS-ID. <i>class-name-1</i> INHERITS <i>class-name-2</i> [METAClass [IS] <i>class-name-3</i>]. [AUTHOR. <i>[comment-entry]</i>] [INSTALLATION. <i>[comment-entry]</i>] [DATE-WRITTEN. <i>[comment-entry]</i>] [DATE-COMPILED. <i>[comment-entry]</i>] [SECURITY. <i>[comment-entry]</i>]	N	N

Paragraphs	Format	Parser	CE
Method Identification Division	<ul style="list-style-type: none"> • Format <IDENTIFICATION ID> DIVISION. METHOD-ID. <i>method-name-1</i> [[IS] [METHOD] OVERRIDE]. [AUTHOR. <i>comment-entry</i>] [INSTALLATION. <i>comment-entry</i>] [DATE-WRITTEN. <i>comment-entry</i>] [DATE-COMPILED. <i>comment-entry</i>] [SECURITY. <i>comment-entry</i>] 	N	N

Environment Division

Input-output section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [[ORGANIZATION [IS]] SEQUENTIAL] [PADDING [CHARACTER] [IS] < <i>data-name-5</i> <i>literal-2</i> >] [RECORD DELIMITER [IS] <STANDARD-1 <i>assignment-name-2</i> >] [ACCESS [MODE] [IS] SEQUENTIAL] [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 2: indexed file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] INDEXED [ACCESS [MODE] [IS] <SEQUENTIAL RANDOM DYNAMIC>] RECORD [KEY] [IS] <i>data-name-2</i> [PASSWORD [IS] <i>data-name-6</i>] ALTERNATE RECORD [KEY] [IS] <i>data-name-3</i> [[WITH] DUPLICATES] [PASSWORD [IS] <i>data-name-7</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 3: relative file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] RELATIVE [ACCESS [MODE] [IS] <SEQUENTIAL [RELATIVE [KEY] [IS] <i>data-name-4</i>] <RANDOM DYNAMIC> RELATIVE [KEY] [IS] <i>data-name-4</i> >] [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 4: line-sequential file-control entries (all platforms except VM) FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[ORGANIZATION [IS]] LINE SEQUENTIAL [ACCESS [MODE] [IS] SEQUENTIAL] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N

Paragraphs and entries	Format	Parser	CE
I-O-CONTROL paragraph	<ul style="list-style-type: none"> • Format 1: sequential I-O I-O-CONTROL. <RERUN ON <assignment-name-1 file-name-1> [EVERY] <integer-1 RECORDS END [OF] <REEL UNIT>> [OF] file-name-1 SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4 MULTIPLE FILE [TAPE] [CONTAINS] file-name-5 [POSITION] integer-2 APPLY WRITE-ONLY [ON] file-name-2>. 	VO	N
	<ul style="list-style-type: none"> • Format 2: relative and indexed I-O I-O-CONTROL. <RERUN ON <assignment-name-1 file-name-1> [EVERY] integer-1 RECORDS [OF] file-name-1 SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4> 	VO	N
	<ul style="list-style-type: none"> • Format 3: line-sequential I-O I-O-CONTROL. SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4. 	VO	N
	<ul style="list-style-type: none"> • Format 4: sort-merge I-O I-O-CONTROL. [RERUN [ON] assignment-name-1] SAME <RECORD SORT SORT-MERGE> [AREA] [FOR] file-name-3 file-name-4. 	VO	N

1-36 Supported COBOL Statements
Supported COBOL for OS/390 statements

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COMPUTER paragraph	<ul style="list-style-type: none"> Format OBJECT-COMPUTER. [<i>computer-name</i> [MEMORY [SIZE] <i>integer</i> <WORDS CHARACTERS MODULES>] [[PROGRAM] [COLLATING] SEQUENCE [IS] <i>alphabet-name</i>] [SEGMENT-LIMIT [IS] <i>priority-number</i>].] 	VO	N
REPOSITORY paragraph	<ul style="list-style-type: none"> Format REPOSITORY. [CLASS <i>class-name-1</i> [[IS] <i>external-class-name-1</i>].] 	N	N
SOURCE-COMPUTER paragraph	<ul style="list-style-type: none"> Format SOURCE-COMPUTER. [<i>computer-name</i> [[WITH] DEBUGGING MODE].] 	VO	N
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> Format SPECIAL-NAMES. [<i>environment-name-1</i> [IS] <i>mnemonic-name-1</i> <i>environment-name-2</i> <[IS] <i>mnemonic-name-2</i> <i>entry-1</i> <i>entry-1</i>>] where <i>entry-1</i> is: <i>entry-1</i>: <ON [STATUS] [IS] <i>condition-1</i> [OFF [STATUS] [IS] <i>conditional-2</i>] OFF [STATUS] [IS] <i>condition-2</i> [ON [STATUS] [IS] <i>conditional-1</i>]> 	V	PS
	[ALPHABET <i>alphabet-name-1</i> [IS] <STANDARD-1 STANDARD-2 NATIVE EBCDIC <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i>]>]	VO	N
	[SYMBOLIC [CHARACTERS] <i>symbolic-character</i> [ARE IS] <i>integer-1</i> [IN] <i>alphabet-name-2</i>]	VO	N
	[CLASS <i>class-name-1</i> [IS] <i>literal-4</i> [<THROUGH THRU> <i>literal-5</i>]	VO	N
	[CURRENCY [SIGN] [IS] <i>literal-6</i> [[WITH] PICTURE SYMBOL <i>literal-7</i>]	VO	N
	[DECIMAL-POINT [IS] COMMA] [.]	V	N

Data Division

Entries and clauses	Format	Parser	CE
Data Division	<ul style="list-style-type: none"> Format: program and method DATA DIVISION DATA DIVISION. [FILE SECTION. [<i>file-description-entry</i> <i>record-description-entry</i>]] [WORKING-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [LOCAL-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [LINKAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] 	V	S
Data Division	<ul style="list-style-type: none"> Format: class DATA DIVISION [WORKING-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] 	N	N
File Description (FD) Entry	<ul style="list-style-type: none"> Format 1: sequential files FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i>] [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]> [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED <i>data-name-2</i>> [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>> [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [RECORDING [MODE] [IS] <i>mode</i>] [CODE-SET [IS] <i>alphabet-name</i>]. 	V	S
	<ul style="list-style-type: none"> Format 2: relative and indexed files FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i>] [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]> [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>> [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>]. 	V	S
	<ul style="list-style-type: none"> Format 3: line-sequential files FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i>] [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]>]. 	V	S

1-38 Supported COBOL Statements
Supported COBOL for OS/390 statements

Entries and clauses	Format	Parser	CE
	<ul style="list-style-type: none"> Format 4: sort/merge files SD <i>file-name-1</i> [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i> [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]]> [DATA <RECORD [IS] RECORDS [ARE]]> <i>data-name-4</i> [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [LABEL <RECORD [IS] RECORDS [ARE]]> <STANDARD OMITTED <i>data-name-2</i>>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [CODE-SET [IS] <i>alphabet-name</i>]. 	V	S
Data Description Entry	<ul style="list-style-type: none"> Format 1 <i>level-number</i> <<i>data-name-1</i> FILLER> [<i>redefines-clause</i>] [<i>blank-when-zero-clause</i>] [<i>external-clause</i>] [<i>global-clause</i>] [<i>justified-clause</i>] [<i>occurs-clause</i>] [<i>picture-clause</i>] [<i>sign-clause</i>] [<i>synchronized-clause</i>] [<i>usage-clause</i>] [<i>value-clause</i>] [<i>data-format-clause</i>] 	V	S
	<ul style="list-style-type: none"> Format 2 66 <i>data-name-1</i> <i>renames-clause</i>. 	V	S
	<ul style="list-style-type: none"> Format 3 88 <i>condition-name</i> <i>value-clause</i>. 	V	S
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> Format BLANK [WHEN] <ZERO ZEROS ZEROES> 	V	S
DATE FORMAT clause	<ul style="list-style-type: none"> Format DATE FORMAT [IS] <i>date-pattern</i> 	VO	N
EXTERNAL clause		V	N
GLOBAL clause		V	N
JUSTIFIED clause	<ul style="list-style-type: none"> Format <JUSTIFIED JUST> [RIGHT] 	VO	N

Entries and clauses	Format	Parser	CE
OCCURS clause	<ul style="list-style-type: none"> Format 1: fixed-length tables OCCURS <i>integer-2</i> [TIMES] [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2: variable-length tables OCCURS <i>integer-1</i> TO <i>integer-2</i> [TIMES] DEPENDING [ON] <i>data-name-1</i> [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
PICTURE clause	<ul style="list-style-type: none"> Format <PICTURE PIC> [IS] <i>character-string</i> 	V	S
REDEFINES clause	<ul style="list-style-type: none"> Format <i>level-number</i> <<i>data-name-1</i> FILLER> REDEFINES <i>data-name-2</i> 	V	S
RENAME clause	<ul style="list-style-type: none"> Format 66 <i>data-name-1</i> RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>] 	V	S
SIGN clause	<ul style="list-style-type: none"> Format SIGN [IS] <LEADING TRAILING>[SEPARATE CHARACTER] 	V	N
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZED SYNC> [LEFT RIGHT] 	VO	N
USAGE clause	<ul style="list-style-type: none"> Format 1: standard USAGE [IS] <BINARY COMP COMP-1 COMP-2 COMP-3 COMP-4 COMP-5 COMPUTATIONAL COMPUTATIONAL-1 COMPUTATIONAL-2 COMPUTATIONAL-3 COMPUTATIONAL-4 COMPUTATIONAL-5 DISPLAY DISPLAY-1 INDEX PACKED-DECIMAL POINTER > 	V	S
	<ul style="list-style-type: none"> Format 2: with 'NATIVE' USAGE [IS] <BINARY NATIVE COMP COMP-1 NATIVE COMP-2 NATIVE COMP-3 COMP-4 NATIVE COMP-5 COMPUTATIONAL COMPUTATIONAL-1 NATIVE COMPUTATIONAL-2 NATIVE COMPUTATIONAL-3 COMPUTATIONAL-4 NATIVE COMPUTATIONAL-5 DISPLAY NATIVE DISPLAY-1 NATIVE INDEX PACKED-DECIMAL POINTER PROCEDURE-POINTER> 	VO	N
	<ul style="list-style-type: none"> Format 3: PROCEDURE-POINTER USAGE [IS] PROCEDURE-POINTER 	VO	N
	<ul style="list-style-type: none"> Format 4: OBJECT REFERENCE USAGE [IS] OBJECT REFERENCE [[METACLASS [OF]] <i>class-name-1</i>] 	N	N

1-40 Supported COBOL Statements
Supported COBOL for OS/390 statements

Entries and clauses	Format	Parser	CE
VALUE clause	<ul style="list-style-type: none"> • Format 1: literal value <i>VALUE [IS] literal</i> 	V	S
	<ul style="list-style-type: none"> • Format 2: condition-name value <i>88 condition-name-1 <VALUE [IS] VALUES [ARE]> literal-1 [<THROUGH THRU> literal-2].</i> 	V	S
	<ul style="list-style-type: none"> • Format 3: NULL value <i>VALUE [IS] <NULL NULLS></i> 	V	S

Procedure Division

Arithmetic operators

Binary operation	Meaning	Format	Parser	CE			
				Gen.	'Read' variables	'Written' variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

Conditional Expressions

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1</i> [IS] [NOT] <NUMERIC ALPHABETIC ALPHABETIC-LOWER ALPHABETIC-UPPER <i>class-name</i> DBCS KANJI> 	V	S	<i>id-1</i>		S
		V O	N			N
Condition-name condition	<ul style="list-style-type: none"> Format condition-name 	V	S	condition-name		S
Relation condition	<ul style="list-style-type: none"> Format 1 <i>operand-1</i> [IS] <[NOT] <GREATER [THAN] LESS [THAN] EQUAL [TO] > GREATER [THAN] OR EQUAL [TO] LESS [THAN] OR EQUAL [TO]> <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>		S
	<ul style="list-style-type: none"> Format 2: pointer data items <ADDRESS OF <i>id-1</i> <i>id-2</i> NULL NULLS> [IS] [NOT] EQUAL [TO] <ADDRESS OF <i>id-3</i> <i>id-4</i> NULL NULLS> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>		N
	<ul style="list-style-type: none"> Format 3: procedure-pointer data items [<i>id-2</i> NULL NULLS] [IS] [NOT] EQUAL [TO] [<i>id-4</i> NULL NULLS] 	V	S	<i>id-2</i> <i>id-4</i>		N
	<ul style="list-style-type: none"> Format 4: object reference data items <<i>object-reference-id-1</i> SELF NULL NULLS> [IS] [NOT] EQUAL [TO] <<i>object-reference-id-2</i> SELF NULL NULLS> 	N	N			N
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	<i>operand-1</i>		S
Switch-status condition	<ul style="list-style-type: none"> Format condition-name 	V	S	condition-name		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT <i>condition-1</i> 	V	S	<i>condition-1</i>		S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1 <AND OR> condition-2</i> 	V	S	<i>condition-1 condition-2</i>		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format <i>relation-condition <AND OR> [NOT] [relational-operator] object</i> 	V	S	<i>relation-condition object</i>		S

Statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format 1: programs and methods (with BY REFERENCE) PROCEDURE DIVISION [USING [[BY] REFERENCE] <i>data-name-1</i>] [RETURNING <i>data-name-2</i>]. 	V	S	<i>data-name-1</i>	<i>data-name-1 data-name-2</i>	N
	<ul style="list-style-type: none"> Format 2: programs and methods (with BY VALUE) PROCEDURE DIVISION [USING [[BY] VALUE] <i>data-name-1</i>] [RETURNING <i>data-name-2</i>]. 	VO	N			N
	<ul style="list-style-type: none"> Format 3: classes PROCEDURE DIVISION. 	N	N			N

1-44 Supported COBOL Statements
Supported COBOL for OS/390 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION structure	<ul style="list-style-type: none"> Format 1: program and method procedure division header [DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES.] <i>section-name</i> SECTION [<i>priority-number</i>]. [<i>paragraph-name</i>. [<i>sentence</i>]] 	V	PS			N
	<ul style="list-style-type: none"> Format 2: classes PROCEDURE DIVISION. [<i>method-definition</i>] 	N	N			N
DECLARATIVES section	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FROM <<i>mnemonic-name</i> <i>environment-name</i>>] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE [YYYYMMDD] DAY [YYYYDDD] DAY-OF-WEEK TIME> 	V	S		<i>id</i>	N
ADD	<ul style="list-style-type: none"> Format 1 ADD <<i>id-1</i> <i>literal</i>> TO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <<i>id-1</i> <i>literal-1</i>> TO <<i>id-2</i> <i>literal-1</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N
ALTER	<ul style="list-style-type: none"> Format ALTER 	V	S			N
	<ul style="list-style-type: none"> <i>procedure-name-1</i> TO [PROCEED TO] <i>procedure-name-2</i> 	VO	N			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
CALL	<ul style="list-style-type: none"> Format 1 CALL <id-1> literal-1> [USING <[[BY] REFERENCE] [ADDRESS OF] id-2 [BY] CONTENT <[ADDRESS OF LENGTH OF] id-3 literal-2>>] [RETURNING id-5] [[[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [ON] OVERFLOW imperative-stmt-3] [END-CALL] 	V	S	id-1 id-2 id-3	id-2	S
	<ul style="list-style-type: none"> Format 2: with OMMITED CALL <id-1> literal-1 procedure-ptr-1> [USING <[[BY] REFERENCE] <[ADDRESS OF] id-2 file-name-1 OMITTED> [BY] CONTENT <[ADDRESS OF LENGTH OF] id-3 literal-2 OMITTED> [BY] VALUE <[ADDRESS OF LENGTH OF] id-4 literal-3>>] [RETURNING id-5] [[[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [ON] OVERFLOW imperative-stmt-3] [END-CALL] 	VO	N			N
	<ul style="list-style-type: none"> Format 3: with file-name-1 CALL <id-1> literal-1 procedure-ptr-1> [USING [[BY] REFERENCE] file-name-1 [RETURNING id-5] [[[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [ON] OVERFLOW imperative-stmt-3] [END-CALL] 	VO	N			N
	<ul style="list-style-type: none"> Format 4: with USING BY VALUE CALL <id-1> literal-1 procedure-ptr-1> [USING [BY] VALUE <[ADDRESS OF LENGTH OF] id-4 literal-3>] [RETURNING id-5] [[[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [ON] OVERFLOW imperative-stmt-3] [END-CALL] 	VO	N			N
	<ul style="list-style-type: none"> Format 5: with procedure-ptr-1 CALL procedure-ptr-1 [USING <[[BY] REFERENCE] <[ADDRESS OF] id-2 file-name-1 OMITTED> [BY] CONTENT <[ADDRESS OF LENGTH OF] id-3 literal-2 OMITTED> [BY] VALUE <[ADDRESS OF LENGTH OF] id-4 literal-3>>] [RETURNING id-5] [[[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [ON] OVERFLOW imperative-stmt-3] [END-CALL] 	VO	N			N
CANCEL	<ul style="list-style-type: none"> Format CANCEL <id-1 literal-1> 	V	S			N

1-46 Supported COBOL Statements
Supported COBOL for OS/390 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> [<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND] [WITH] <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files CLOSE <i>file-name-1</i> [[WITH] LOCK] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 3: line-sequential files CLOSE <i>filename-1</i> [<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND] [WITH] <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <i>id-1</i> [ROUNDED] [EQUAL] <i>arithmetic-expr</i> [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-COMPUTE] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S
CONTINUE	<ul style="list-style-type: none"> Format CONTINUE 	V	S			N
DELETE	<ul style="list-style-type: none"> Format DELETE <i>file-name-1</i> [RECORD] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-DELETE] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISPLAY	<ul style="list-style-type: none"> Format DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <<i>mnemonic-name-1</i> <i>environment-name-1</i>>] [[WITH] NO ADVANCING] 	V	S	<i>id-1</i>		S
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <<i>id-1</i> <i>literal-1</i>> INTO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-DIVIDE] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
ENTRY	<ul style="list-style-type: none"> Format 1 ENTRY literal-1 [USING [[BY] REFERENCE] id-1]] [BY] VALUE id-1] 	V	S	id-1		S
	<ul style="list-style-type: none"> Format 2 ENTRY literal-1 [USING [[BY] VALUE] id-1] 	VO	N			S

1-48 Supported COBOL Statements
Supported COBOL for OS/390 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
EVALUATE	<ul style="list-style-type: none"> Format EVALUATE <id-1 literal-1 expr-1 TRUE FALSE> [ALSO <id-2 literal-2 expr-2 TRUE FALSE>] WHEN phrase-1 [ALSO phrase-2] imperative-stmt-1 [WHEN OTHER imperative-stmt-2] [END-EVALUATE] Phrases phrase-1 and phrase-2 should be represented in the following form: <ANY condition-1 TRUE FALSE [NOT] <id-3 literal-1 arithmetic-expr-1> [<THROUGH THRU> <id-4 literal-2 arithmetic-expr-2>]> 	V	S	id-1 expr-1 id-2 expr-2 condition-1 id-3 id-4 arithmetic-expr1 arithmetic-expr2		N
EXIT	<ul style="list-style-type: none"> Format paragraph-name. EXIT. 	V	S			N
EXIT METHOD	<ul style="list-style-type: none"> Format EXIT METHOD. 	N	N			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
GO-BACK	<ul style="list-style-type: none"> Format GOBACK 	V	S			N
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO [TO] procedure-name-1 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO [TO] procedure-name-1 DEPENDING [ON] id-1 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3: altered paragraph-name. GO [TO]. 	V	S			N
	<ul style="list-style-type: none"> Format 4: MORE-LABELS GO [TO] MORE-LABELS 	N	N			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
IF	<ul style="list-style-type: none"> Format IF <i>condition-1</i> THEN <stmt-1 NEXT SENTENCE> [ELSE <stmt-2 NEXT SENTENCE>] [END-IF] 	V	S	<i>condition-1</i>		N
INITIALIZE	<ul style="list-style-type: none"> Format INITIALIZE <i>id-1</i> [REPLACING <ALPHABETIC ALPHANUMERIC NUMERIC ALPHANUMERIC-EDITED NUMERIC-EDITED DBCS EGCS> [DATA] BY <<i>id-2</i> <i>literal-1</i>>] 	V	S	<i>id-2</i>	<i>id-1</i>	N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-5</i> <i>id-4</i> <i>id-3</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 4 INSPECT <i>id-1</i> CONVERTING <<i>id-6</i> <i>literal-4</i>> TO <<i>id-7</i> <i>literal-5</i>> [BEFORE AFTER] [INITIAL] <<i>id-4</i> <i>literal-2</i>> 	V	S	<i>id-1</i> <i>id-6</i> <i>id-7</i> <i>id-4</i>	<i>id-1</i>	N
INVOKE	<ul style="list-style-type: none"> Format INVOKE <<i>id-1</i> <i>class-name-1</i> SELF [<i>class-name-2</i> OF] SUPER> <<i>literal-1</i> <i>id-2</i>> [USING <[[BY] REFERENCE] <[ADDRESS OF] <i>id-3</i> OMITTED> [[BY] CONTENT] <[ADDRESS OF LENGTH OF] <i>id-4</i> <i>literal-2</i> OMITTED> [[BY] VALUE] <[ADDRESS OF LENGTH OF] <i>id-5</i> <i>literal-3</i>>>] [RETURNING <i>id-6</i>] [[ON] EXCEPTION <i>imperative-stmt-1</i>] [NOT [ON] EXCEPTION <i>imperative-stmt-2</i>] [END-INVOKE] 	N	N			N

1-50 Supported COBOL Statements
Supported COBOL for OS/390 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> 	V	S	<i>file-name-1</i> <i>data-name-1</i>	<i>file-name-4</i>	N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N	<i>file-name-2</i> <i>file-name-3</i>		N
	USING <i>file-name-2</i> <i>file-name-3</i> <OUTPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i> >	V	S			N
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <<i>id-1</i> <i>literal-1</i>> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	N
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <<i>id-1</i> <i>literal-1</i>> BY <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-MULTIPLY] 	V	S	<i>id-1</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <<i>id-1</i> <i>literal-1</i>> BY <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-MULTIPLY] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
OPEN	<ul style="list-style-type: none"> Format 1: sequential files OPEN <INPUT <i>file-name-1</i> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N
	[REVERSED WITH NO REWIND]	V	N			N
	OUTPUT <i>file-name-2</i>	V	S			N
	[WITH NO REWIND]	V	N			N
	I-O <i>file-name-3</i> EXTEND <i>file-name-4</i> >	V	S			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files OPEN <INPUT <i>file-name-1</i> OUTPUT <i>file-name-2</i> I-O <i>file-name-3</i> EXTEND <i>file-name-4</i>> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N
<ul style="list-style-type: none"> Format 3: line-sequential files OPEN <INPUT <i>file-name-1</i> OUTPUT <i>file-name-2</i> EXTEND <i>file-name-4</i>> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N	
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM <<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <i>imperative-stmt-1</i> END-PERFORM> 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM <<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <<i>id-1</i> <i>integer-1</i>> TIMES <i>imperative-stmt-1</i> END-PERFORM> 	V	S	<i>id-1</i>		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] UNTIL <i>condition-1</i> <i>imperative-stmt-1</i> END-PERFORM 	V	S	<i>condition-1</i>		N
	<ul style="list-style-type: none"> Format 4: with UNTIL phrase, without END-PERFORM PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] [[WITH] TEST <BEFORE AFTER>] UNTIL <i>condition-1</i> 	S	S	<i>condition-1</i>		N

1-52 Supported COBOL Statements
Supported COBOL for OS/390 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PERFORM	<ul style="list-style-type: none"> Format 5: with VARYING phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	id-3 id-4 condition-1	id-2	N
	<ul style="list-style-type: none"> Format 6: with VARYING phrase, without END-PERFORM PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 [AFTER <id-5 index-name-3> FROM <id-6 index-name-4 literal-3> BY <id-7 literal-4> UNTIL condition-2] <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-2 	N	N			N
READ	<ul style="list-style-type: none"> Format 1: sequential retrieval READ file-name-1 	V	S	file-name-1	file-name-1 id-1	N
	[NEXT PREVIOUS] [RECORD]	VO	N			N
	[INTO id-1] [[AT] END imperative-stmt-1] [NOT [AT] END imperative-stmt-2] [END-READ]	V	S			S
	<ul style="list-style-type: none"> Format 2: random retrieval READ file-name-1 [RECORD] [INTO id-1] [KEY [IS] data-name-1] [INVALID [KEY] imperative-stmt-3] [NOT INVALID [KEY] imperative-stmt-4] [END-READ] 	V	S	file-name-1 data-name-1	file-name-1 id-1	N
RELEASE	<ul style="list-style-type: none"> Format RELEASE record-name-1 [FROM id-1] 	V	S	record-name-1 id-1	record-name-1	N
RETURN	<ul style="list-style-type: none"> Format RETURN file-name-1 [RECORD] [INTO id-1] [[AT] END imperative-stmt-1] [NOT [AT] END imperative-stmt-2] [END-RETURN] 	V	S	file-name-1	id-1	N
RE-WRITE	<ul style="list-style-type: none"> Format REWRITE file-name-1 [FROM id-1] [INVALID [KEY] imperative-stmt-1] [NOT INVALID [KEY] imperative-stmt-2] [END-REWRITE] 	V	S	file-name-1 id-1	file-name-1	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [[AT] END <i>imperative-stmt-1</i>] WHEN <i>condition-1</i> <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>condition-1</i>	<i>id-2</i>	N
	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [[AT] END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> [IS] EQUAL [TO] <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> [IS] EQUAL [TO] <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>data-name-1</i>		N
SET	<ul style="list-style-type: none"> Format 1: basic table handling SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-2</i> <i>integer-1</i>> 	V	S	<i>index-name-2</i> <i>id-2</i>	<i>index-name-1</i> <i>id-1</i>	N
	<ul style="list-style-type: none"> Format 2: adjusting indexes SET <i>index-name-3</i> <UP BY DOWN BY> <<i>id-3</i> <i>integer-2</i>> 	V	S	<i>index-name-3</i> <i>id-3</i>	<i>index-name-3</i>	N
	<ul style="list-style-type: none"> Format 3: external switches SET <i>mnemonic-name-1</i> TO <ON OFF> 	V	S		<i>mnemonic-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: condition-names SET <i>condition-name-1</i> TO TRUE 	V	S		<i>condition-variable</i>	S
	<ul style="list-style-type: none"> Format 5: USAGE IS POINTER data items SET <<i>id-4</i> ADDRESS OF <i>id-5</i>> TO <<i>id-6</i> ADDRESS OF <i>id-7</i> NULL NULLS> 	V	S	<i>id-6</i> <i>id-7</i>	<i>id-4</i> <i>id-5</i>	N
	<ul style="list-style-type: none"> Format 6: USAGE IS PROCEDURE-POINTER data items SET <i>procedure-pointer-data-item-1</i> TO <<i>procedure-pointer-data-item-2</i> ENTRY <<i>id-8</i> <i>literal-1</i>> NULL NULLS <i>pointer-data-item-3</i>> 	VO	N			N
	<ul style="list-style-type: none"> Format 7: USAGE OBJECT REFERENCE data items SET <i>object-reference-id-1</i> TO <<i>object-reference-id-2</i> NULL SELF> 	N	N			N

1-54 Supported COBOL Statements
Supported COBOL for OS/390 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> [[WITH] DUPLICATES [IN] [ORDER]] 	V	S	<i>file-name-1</i> <i>data-name-1</i> <i>file-name-2</i>	<i>file-name-3</i>	N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N			N
	<USING <i>file-name-2</i> INPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <GIVING <i>file-name-3</i> OUTPUT PROCEDURE [IS] <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i>]>	V	S			N
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [KEY [IS] <EQUAL [TO] = LESS [THAN] < GREATER [THAN] > NOT LESS [THAN] NOT < NOT GREATER [THAN] NOT > LESS [THAN] OR EQUAL [TO] <= GREATER [THAN] OR EQUAL [TO] >=] <i>data-name-1</i> [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-START] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N
STRING	<ul style="list-style-type: none"> Format STRING <<i>id-1</i> <i>literal-1</i>> DELIMITED [BY] <<i>id-2</i> <i>literal-2</i> SIZE> INTO <i>id-3</i> [[WITH] POINTER <i>id-4</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-STRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>	<i>id-3</i> <i>id-4</i>	S
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <<i>id-1</i> <i>literal-1</i>> FROM <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <<i>id-1</i> <i>literal-1</i>> FROM <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> <i>id-1</i> FROM <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING <i>id-1</i> [DELIMITED [BY] [ALL] <<i>id-2</i> <i>literal-1</i>> [OR [ALL] <<i>id-3</i> <i>literal-2</i>>]] INTO <i>id-4</i> [DELIMITER [IN] <i>id-5</i>] [COUNT [IN] <i>id-6</i>] [[WITH] POINTER <i>id-7</i>] [TALLYING [IN] <i>id-8</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-UNSTRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-7</i> <i>id-8</i>	<i>id-4</i> <i>id-5</i> <i>id-6</i> <i>id-7</i> <i>id-8</i>	S
WRITE	<ul style="list-style-type: none"> Format 1: sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: sequential files with 'BEFORE/AFTER' WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [<<BEFORE AFTER> [ADVANCING] <<<i>id-1</i> <i>integer-1</i>> [LINE LINES] <i>mnemonic-name-1</i> PAGE>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	VO	N			N
	<ul style="list-style-type: none"> Format 3: indexed and relative files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: line-sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 5: line-sequential files with 'AFTER' WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [AFTER [ADVANCING] <<i>id-2</i> <i>integer-1</i>> [LINE LINES] PAGE>] [END-WRITE] 	VO	N			N

Intrinsic functions

Functions	Parser	CE	
		Gen.	DB
ACOS	V	S	N
ANNUITY	V	S	N
ASIN	V	S	N
ATAN	V	S	N
CHAR	V	S	N
COS	V	S	N
CURRENT-DATE	V	S	N
DATE-OF-INTEGER	V	S	N
DATE-TO-YYYYMMDD	V	S	N
DATEVAL	V	S	N
DAY-OF-INTEGER	V	S	N
DAY-TO-YYYYDDD	V	S	N
FACTORIAL	V	S	N
INTEGER	V	S	N
INTEGER-OF-DATE	V	S	N
INTEGER-OF-DAY	V	S	N
INTEGER-PART	V	S	N
LENGTH	V	S	N
LOG	V	S	N
LOG10	V	S	N
LOWER-CASE	V	S	N

Functions	Parser	CE	
		Gen.	DB
MAX	V	S	N
MEAN	V	S	N
MEDIAN	V	S	N
MIDRANGE	V	S	N
MIN	V	S	N
MOD	V	S	N
NUMVAL	V	S	N
NUMVAL-C	V	S	N
ORD	V	S	N
ORD-MAX	V	S	N
ORD-MIN	V	S	N
PRESENT-VALUE	V	S	N
RANDOM	V	S	N
RANGE	V	S	N
REM	V	S	N
REVERSE	V	S	N
SIN	V	S	N
SQRT	V	S	N
STANDARD-DEVIATION	V	S	N
SUM	V	S	N
TAN	V	S	N
UNDATE	V	S	N

1-58 Supported COBOL Statements
Supported COBOL for OS/390 statements

Functions	Parser	CE	
		Gen.	DB
UPPER-CASE	V	S	N
VARIANCE	V	S	N
WHEN-COMPILED	V	S	N
YEAR-TO-YYYY	V	S	N
YEARWINDOW	V	S	N

Compile-directing statements and directives

Statements and directives	Format	Parser	CE
BASIS	<ul style="list-style-type: none"> Format [sequence-number] BASIS <basis-name literal-1> 	VO	N
CBL (PROCESS)	<ul style="list-style-type: none"> Format <CBL PROCESS> [options-list] 	VO	N
CALLINTERFACE directive	<ul style="list-style-type: none"> Format < >>CALLINTERFACE >>CALLINT > [SYSTEM OPTLINK FAR16 PASCAL16 CDECL] [DESC DESCRIPTOR NODESC NODESCRIPTOR] 	VO	N
*CONTROL (*CBL)	<ul style="list-style-type: none"> Format <*CONTROL *CBL> <SOURCE NOSOURCE LIST NOLIST MAP NOMAP> 	VO	N
COPY	<ul style="list-style-type: none"> Format COPY <text-name-1 literal-1> [<OF IN> <library-name literal-2>] [SUPPRESS] [REPLACING operand-1 BY operand-2] 	V	S
DELETE	<ul style="list-style-type: none"> Format [sequence-number] DELETE sequence-number-field 	VO	N
EJECT	<ul style="list-style-type: none"> Format EJECT [.] 	VO	N

Statements and directives	Format	Parser	CE
ENTER	<ul style="list-style-type: none"> Format ENTER <i>language-name-1</i> [<i>routine-name-1</i>]. 	VO	N
INSERT	<ul style="list-style-type: none"> Format [<i>sequence-number</i>] INSERT <i>sequence-number-field</i> 	VO	N
READY or RESET TRACE	<ul style="list-style-type: none"> Format <READY RESET> TRACE. 	VO	N
REPLACE	<ul style="list-style-type: none"> Format 1 REPLACE ==<i>pseudo-text-1</i>== BY ==<i>pseudo-text-2</i>==. 	V	S
	<ul style="list-style-type: none"> Format 2 REPLACE OFF. 	V	S
SERVICE LABEL	<ul style="list-style-type: none"> Format SERVICE LABEL 	VO	N
SERVICE RELOAD	<ul style="list-style-type: none"> Format SERVICE RELOAD <i>id-1</i> 	VO	N
SKIP1/2/3	<ul style="list-style-type: none"> Format <SKIP1 SKIP2 SKIP3>. 	VO	N
TITLE	<ul style="list-style-type: none"> Format TITLE <i>literal</i>. 	VO	N
USE	<ul style="list-style-type: none"> Format 1: EXCEPTION ERROR declarative USE [GLOBAL] AFTER [STANDARD] <EXCEPTION ERROR> PROCEDURE [ON] <<i>file-name-1</i> INPUT OUTPUT I-O EXTEND> 	VO	N
	<ul style="list-style-type: none"> Format 2: LABEL declarative USE [GLOBAL] AFTER [STANDARD] [BEGINNING ENDING] [FILE REEL UNIT] LABEL PROCEDURE [ON] <<i>file-name-1</i> INPUT OUTPUT I-O EXTEND> 	VO	N

Supported COBOL/400 statements

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S
NULL / NULLS	V	S

Special registers

Registers	Parser	CE
ADDRESS OF	V	PS
DB-FORMAT-NAME	N	N
DEBUG -ITEM	N	N
LENGTH OF	V	S
LINAGE-COUNTER	V	N
WHEN COMPILED	V	N

Literals

Literal type	Format	Parser	CE
Boolean literals	<ul style="list-style-type: none"> Format Example: B'0' 	V	N
Nonnumeric literal	<ul style="list-style-type: none"> Format 1: with quotes (can contain any allowable character from the EBCDIC character set) Example: "THIS ISN'T WRONG" 	V	S
	<ul style="list-style-type: none"> Format 2: hexadecimal notation X"hexadecimal-digits" 	V	S
Numeric literals	<ul style="list-style-type: none"> Format: fixed-point Example: -2.71828 	V	S
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format <i>text-name-1</i> [<IN OF> <i>library-name-1</i>] 	V	S
to Procedure Division	<ul style="list-style-type: none"> Format 1 <i>paragraph-name-1</i> [<IN OF> <i>section-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2 <i>section-name-1</i> 	V	S

1-62 Supported COBOL Statements
Supported COBOL/400 statements

Division	Format	Parser	CE
to Data Division	<ul style="list-style-type: none"> • Format 1: simple data reference <i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> • Format 2 <condition-name-1 data-name-1> [<IN OF> data-name-2] 	V	S
	<ul style="list-style-type: none"> • Format 3 LINAGE-COUNTER [<IN OF> file-name-2] 	V	S
Condition names	<ul style="list-style-type: none"> • Format 1: Data Division <i>condition-name-1</i> [<IN OF> data-name-1] [<IN OF> file-name-1] (<i>subscript</i>) 	V	S
	<ul style="list-style-type: none"> • Format 2: Special-Names paragraph <i>condition-name-1</i> [<IN OF> <i>mnemonic-name-1</i>] 	V	S
Subscript	<ul style="list-style-type: none"> • Format <<i>integer-1</i> ALL data-name-3 [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>]> 	V	S
Reference modification	<ul style="list-style-type: none"> • Format <<i>data-name-1</i>> [<IN OF> data-name-2] (<i>leftmost-character-position: [length]</i>) 	V	S

COBOL program structure

Program type	Format	Parser	CE
Single program	<ul style="list-style-type: none"> • Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> END-PROGRAM <i>program-name-1</i>. 	V	S

Program type	Format	Parser	CE
Nested program	<ul style="list-style-type: none"> • Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> <i>nested source program</i> END-PROGRAM <i>program-name-1</i>. 	N	N

Identification Division

Paragraphs	Format	Parser	CE
Program Identification Division	<ul style="list-style-type: none"> • Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name</i> [[IS] <RECURSIVE COMMON [INITIAL] INITIAL [COMMON]> [PROGRAM]]. [AUTHOR. <i>comment-entry</i>] [INSTALLATION. <i>comment-entry</i>] [DATE-WRITTEN. <i>comment-entry</i>] [DATE-COMPILED. <i>comment-entry</i>] [SECURITY. <i>comment-entry</i>] 	V	S

1-64 Supported COBOL Statements
Supported COBOL/400 statements

Environment Division

Input-output section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> <i>literal-1</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [[ORGANIZATION [IS]] SEQUENTIAL] [PADDING [CHARACTER] [IS] < <i>data-name-5</i> <i>literal-2</i> >] [RECORD DELIMITER [IS] <STANDARD-1 <i>assignment-name-2</i> >] [ACCESS [MODE] [IS] SEQUENTIAL] [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 2: indexed file-control entries FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> <i>literal-1</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] INDEXED [ACCESS [MODE] [IS] <SEQUENTIAL RANDOM DYNAMIC->] RECORD [KEY] [IS] < <i>data-name-2</i> EXTERNALLY-SCRIBED-KEY> [PASSWORD [IS] <i>data-name-6</i>] ALTERNATE RECORD [KEY] [IS] <i>data-name-3</i> [PASSWORD [IS] <i>data-name-7</i>] [[WITH] DUPLICATES] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 3: relative file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> <i>literal-1</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] RELATIVE [ACCESS [MODE] [IS] <SEQUENTIAL [RELATIVE [KEY] [IS] <i>data-name-4</i>] <RANDOM DYNAMIC->] RELATIVE [KEY] [IS] <i>data-name-4</i> > [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 4: sort or merge FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> <i>literal-1</i> >	V	S
	<ul style="list-style-type: none"> Format 5: transaction FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> <i>literal-1</i> > [[ORGANIZATION [IS]] TRANSACTION] [ACCESS [MODE] [IS] <SEQUENTIAL DYNAMIC RELATIVE [KEY] [IS] <i>data-name-1</i> > [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-2</i> [<i>data-name-3</i>] [CONTROL-AREA [IS] <i>data-name-4</i>].	VO	N

1-66 Supported COBOL Statements
Supported COBOL/400 statements

Paragraphs and entries	Format	Parser	CE
I-O-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential files I-O I-O-CONTROL. [RERUN ON <assignment-name-1 file-name-1> [EVERY] integer-1 RECORDS [OF] file-name-2] [SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4] [MULTIPLE FILE [TAPE] [CONTAINS] file-name-5 [POSITION integer-2]] [COMMITMENT CONTROL [FOR] file-name-6]. 	VO	N
	<ul style="list-style-type: none"> Format 2: indexed and relative files I-O I-O-CONTROL. [RERUN ON <assignment-name-1 file-name-1> [EVERY] integer-1 RECORDS [OF] file-name-2] [SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4] [COMMITMENT CONTROL [FOR] file-name-6]. 	VO	N
	<ul style="list-style-type: none"> Format 3: sort-merge I-O I-O-CONTROL. SAME <RECORD SORT SORT-MERGE> [AREA] [FOR] file-name-3 file-name-4. 	VO	N

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COM-PUTER paragraph	<ul style="list-style-type: none"> Format OBJECT-COMPUTER. [computer-name [MEMORY [SIZE] integer <WORDS CHARACTERS MODULES>] [[PROGRAM] [COLLATING] SEQUENCE [IS] alphabet-name] [SEGMENT-LIMIT [IS] priority-number].] 	VO	N
SOURCE-COM-PUTER paragraph	<ul style="list-style-type: none"> Format SOURCE-COMPUTER. [computer-name [[WITH] DEBUGGING MODE].] 	VO	N

Paragraphs and entries	Format	Parser	CE
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> Format <p>SPECIAL-NAMES. [<i>environment-name-1</i> [IS] <i>mnemonic-name-1</i> <i>environment-name-2</i> <[IS] <i>mnemonic-name-2</i> <i>entry-1</i> <i>entry-1</i>>] where <i>entry-1</i> is: <i>entry-1</i>: <ON [STATUS] [IS] <i>condition-1</i> [OFF [STATUS] [IS] <i>conditional-2</i>] OFF [STATUS] [IS] <i>condition-2</i> [ON [STATUS] [IS] <i>conditional-1</i>]></p>	V	PS
	[ALPHABET <i>alphabet-name-1</i> [IS] <STANDARD-1 STANDARD-2 NATIVE EBCDIC <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i>]>]	VO	N
	[SYMBOLIC [CHARACTERS] <i>symbolic-character</i> [ARE IS] <i>integer-1</i> [IN] <i>alphabet-name-2</i>]	VO	N
	[CLASS <i>class-name-1</i> [IS] <i>literal-4</i> [<THROUGH THRU> <i>literal-5</i>]]	VO	N
	[CURRENCY [SIGN] [IS] <i>literal-6</i> [[WITH] PICTURE SYMBOL <i>literal-7</i>]]	VO	N
	[DECIMAL-POINT [IS] COMMA]	V	S
	<CONSOLE [IS] CRT CURSOR [IS] <i>data-name-1</i> CRT STATUS [IS] <i>data-name-2</i> > [.]	V	N

Data Division

Entries and clauses	Format	Parser	CE
Data Division	<ul style="list-style-type: none"> Format: program and method DATA DIVISION <p>DATA DIVISION. [FILE SECTION. [<i>file-description-entry</i> <i>record-description-entry</i>]] [WORKING-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [LINKAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]]</p>	V	S
	<ul style="list-style-type: none"> Format 2: relative and indexed files <p>FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i>] [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>].</p>	V	S

1-68 Supported COBOL Statements
Supported COBOL/400 statements

Entries and clauses	Format	Parser	CE
File Description (FD) Entry	<ul style="list-style-type: none"> Format 1: Formatfile, Database, Disk FD <i>file-name-1</i> [BLOCK [CONTAINS] <i>integer-1</i> TO <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>]. 	V	S
	<ul style="list-style-type: none"> Format 2: Diskette, Tapefile FD <i>file-name-1</i> [BLOCK [CONTAINS] <i>integer-1</i> TO <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [CODE-SET [IS] <i>alphabet-name</i>]. 	V	S
	<ul style="list-style-type: none"> Format 3: Printer FD <i>file-name-1</i> [BLOCK [CONTAINS] <i>integer-1</i> TO <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [CODE-SET [IS] <i>alphabet-name</i>]. 	V	S
	<ul style="list-style-type: none"> Format 4: sort/merge SD <i>file-name-1</i> [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS]>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>]. 	V	S
	<ul style="list-style-type: none"> Format 5: transaction FD <i>file-name-1</i> [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>]. 	V	S

Entries and clauses	Format	Parser	CE
Data Description Entry	<ul style="list-style-type: none"> Format 1 <i>level-number <data-name-1 FILLER></i> <i>[redefines-clause like-clause] [blank-when-zero-clause] [justified-clause] [occurs-clause] [picture-clause] [sign-clause] [synchronized-clause] [usage-clause] [value-clause]</i> 	V	S
	<ul style="list-style-type: none"> Format 2 <i>66 data-name-1 renames-clause.</i> 	V	S
	<ul style="list-style-type: none"> Format 3 <i>88 condition-name value-clause.</i> 	V	S
	<ul style="list-style-type: none"> Format 4: Boolean data <i>level-number <data-name-1 FILLER></i> <i>[redefines-clause like-clause] [picture-clause] [usage-clause] [occurs-clause] [<INDICATOR INDICATORS INDIC> integer-1] [synchronized-clause] [justified-clause] [value-clause]</i> 	V	S
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> Format <i>BLANK [WHEN] ZERO</i> 	V	S
DATE FORMAT clause	<ul style="list-style-type: none"> Format <i>DATE FORMAT [IS] date-pattern</i> 	VO	N
EXTERNAL clause		V	N
GLOBAL clause		V	N
JUSTIFIED clause	<ul style="list-style-type: none"> Format <i><JUSTIFIED JUST> [RIGHT]</i> 	VO	N
LIKE clause	<ul style="list-style-type: none"> Format <i>LIKE data-name [(integer)]</i> 	V	S
OCCURS clause	<ul style="list-style-type: none"> Format 1: fixed-length tables <i>OCCURS integer-2 [TIMES] [<ASCENDING DESCENDING> [KEY] [IS] data-name-2] [INDEXED [BY] index-name-1]</i> 	V	S
	<ul style="list-style-type: none"> Format 2: variable-length tables <i>OCCURS integer-1 TO integer-2 [TIMES] DEPENDING [ON] data-name-1 [<ASCENDING DESCENDING> [KEY] [IS] data-name-2] [INDEXED [BY] index-name-1]</i> 	V	S

1-70 Supported COBOL Statements
Supported COBOL/400 statements

Entries and clauses	Format	Parser	CE
PICTURE clause	<ul style="list-style-type: none"> Format <PICTURE PIC> [IS] <i>character-string</i> 	V	S
REDEFINES clause	<ul style="list-style-type: none"> Format <i>level-number</i> <<i>data-name-1</i> FILLER> REDEFINES <i>data-name-2</i> 	V	S
RENAMES clause	<ul style="list-style-type: none"> Format 66 <i>data-name-1</i> RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>] 	V	S
SIGN clause	<ul style="list-style-type: none"> Format SIGN [IS] <LEADING TRAILING>[SEPARATE CHARACTER] 	V	N
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZED SYNC> [LEFT RIGHT] 	VO	N
USAGE clause	<ul style="list-style-type: none"> Format USAGE [IS] <BINARY COMP COMP-3 COMP-4 COMPUTATIONAL COMPUTATIONAL-3 COMPUTATIONAL-4 DISPLAY INDEX PACKED-DECIMAL POINTER > 	V	S
VALUE clause	<ul style="list-style-type: none"> Format 1: literal value VALUE [IS] <i>literal</i> 	V	S
	<ul style="list-style-type: none"> Format 2: condition-name value 88 <i>condition-name-1</i> <VALUE [IS] VALUES [ARE]> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]. 	V	S
	<ul style="list-style-type: none"> Format 3: NULL value VALUE [IS] <NULL NULLS> 	V	S

Procedure Division

Arithmetic operators

Operation	Meaning	Format	Parser	CE			
				Gen.	Read variables	Written variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

1-72 Supported COBOL Statements
Supported COBOL/400 statements

Conditional Expressions

Conditions	Format	Parser	CE			
			Gen.	Read variables	Written variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1</i> [IS] [NOT] <NUMERIC ALPHABETIC ALPHABETIC-LOWER ALPHABETIC-UPPER <i>class-name</i>> 	V	S	<i>id-1</i>		S
		VO	N			N
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] <[NOT] <GREATER [THAN] LESS [THAN] EQUAL [TO] > GREATER [THAN] OR EQUAL [TO] LESS [THAN] OR EQUAL [TO]> <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>		S
	<ul style="list-style-type: none"> Format 2: ADDRESS comparison <ADDRESS OF <i>id-1</i> <i>id-2</i> NULL NULLS> [IS] [NOT] <EQUAL [TO] => <ADDRESS OF <i>id-3</i> <i>id-4</i> NULL NULLS> 	V	N			N
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	<i>operand-1</i>		S
Switch-status condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT <i>condition-1</i> 	V	S	<i>condition-1</i>		S
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1</i> <AND OR> <i>condition-2</i> 	V	S	<i>condition-1</i> <i>condition-2</i>		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format <i>relation-condition</i> <AND OR> [NOT] [<i>relational-operator</i>] <i>object</i> 	V	S	<i>relation-condition</i> <i>object</i>		S

Statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format PROCEDURE DIVISION [USING <i>data-name-1</i>]. 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N
PROCEDURE DIVISION structure	<ul style="list-style-type: none"> Format: procedure division header [DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES.] <i>section-name</i> SECTION [<i>priority-number</i>]. [<i>paragraph-name</i>. [<i>sentence</i>]] 	V	S			S
DECLARATIVES section	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES. 	V	PS			N

1-74 Supported COBOL Statements
Supported COBOL/400 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FROM <<i>mnemonic-name</i> <i>environment-name</i>>] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE DAY TIME> 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 3: feedback, local data area ACCEPT <i>id</i> FROM <i>mnemonic-name</i> [FOR <<i>file-name</i> <i>id-2</i> <i>literal</i>>] 	V	S	<i>id-2</i>	<i>id</i>	
	<ul style="list-style-type: none"> Format 4: PIP data area ACCEPT <i>id</i> FROM <i>mnemonic-name</i> [[ON] EXCEPTION <i>imperative-statement-1</i>] [NOT [ON] EXCEPTION <i>imperative-statement-1</i>] [END-ACCEPT] 	V	S		<i>id</i>	
	<ul style="list-style-type: none"> Format 5: attribute data ACCEPT <i>id</i> FROM <i>mnemonic-name</i> [FOR <<i>id-2</i> <i>literal</i>> [FOR <i>file-name-1</i>]] 	V				
	<ul style="list-style-type: none"> Format 6: workstation I/O ACCEPT <i>id</i> FROM <i>mnemonic-name</i> [FROM CRT MODE [IS] BLOCK [AT] <LINE [NUMBER] <<i>id-1</i> <i>integer-1</i>> <COLUMN COL> [NUMBER] <<i>id-2</i> <i>integer-2</i>>> [AT] <<i>id-3</i> <i>integer-3</i>> WITH <<AUTO AUTO-SKIP> <BELL BEEP> BLINK <FULL LENGTH-CHECK> HIGHLIGHT <REQUIRED EMPTY-CHECK> REVERSE-VIDEO <SECURE NO-ECHO> UNDERLINE LEFT-JUSTIFY RIGHT-JUSTIFY SPACE-FILL TRAILING-SIGN UPDATE ZERO-FILL SIZE [IS] <<i>id-4</i> <i>integer-4</i>> PROMPT [CHARACTER IS] <<i>id-5</i> <i>literal</i>> FOREGROUND-COLOR [IS] <<i>integer-5</i> BACKGROUND-COLOR [IS] <<i>integer-6</i>> [[ON] EXCEPTION <i>imperative-statement-1</i>] [NOT [ON] EXCEPTION <i>imperative-statement-1</i>] [END-ACCEPT] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i>	<i>id</i>	
ACQUIRE	<ul style="list-style-type: none"> Format ACQUIRE <<i>id</i> <i>literal</i>> FOR <i>file-name</i> 	V	S		<i>id</i>	

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ADD	<ul style="list-style-type: none"> Format 1 ADD <id-1 literal> TO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <id-1 literal-1> TO <id-2 literal-1> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> id-1 TO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-2	N
ALTER	<ul style="list-style-type: none"> Format ALTER procedure-name-1 TO [PROCEED TO] procedure-name-2 	V	S			N
CALL	<ul style="list-style-type: none"> Format 1: with ON OVERFLOW CALL <id-1 literal-1> [USING <[[BY] REFERENCE] <[ADDRESS OF] id-2 file-name-1> [BY] CONTENT <[ADDRESS OF LENGTH OF] id-3 literal-2>>] [ON] OVERFLOW imperative-stmt-1] [END-CALL] 	V	S	id-1 id-2 id-3	id-2	S
	<ul style="list-style-type: none"> Format 2: with ON EXCEPTION CALL <id-1 literal-1> [USING <[[BY] REFERENCE] <[ADDRESS OF] id-2 id-3 file-name-1> [BY] CONTENT <[ADDRESS OF LENGTH OF] id-3 literal-2>>] [[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [END-CALL] 	V	S	id-1 id-2 id-3	id-2	S
	<ul style="list-style-type: none"> Format 3: graphics support CALL "GDDM" USING routine-name [data-name-1] 	V	N			
CANCEL	<ul style="list-style-type: none"> Format CANCEL <id-1 literal-1> 	V	S			N

1-76 Supported COBOL Statements
Supported COBOL/400 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	[<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND] [WITH] <NO REWIND LOCK>]	VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files, transaction CLOSE <i>file-name-1</i> 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	[[WITH] LOCK]	VO	N			N
COMMIT	<ul style="list-style-type: none"> Format COMMIT 	V	S			
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <i>id-1</i> [ROUNDED] = <i>arithmetic-expr</i> [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-COMPUTE] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S
CONTINUE	<ul style="list-style-type: none"> Format CONTINUE 	V	S			N
DELETE	<ul style="list-style-type: none"> Format DELETE <i>file-name-1</i> [RECORD] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-DELETE] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISPLAY	<ul style="list-style-type: none"> Format 1: data transfer DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <<i>mnemonic-name-1</i> <i>environment-name-1</i>>] 	V	S	<i>id-1</i>		S
	<ul style="list-style-type: none"> Format 2: local data area DISPLAY <<i>id-1</i> <i>literal-1</i>> UPON <i>mnemonic-name-1</i> [[FOR] <<i>id-2</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-2</i>		
	<ul style="list-style-type: none"> Format 3: workstation I/O DISPLAY <<<i>id-1</i> <i>literal-1</i>> UPON <CRT CRT-UNDER> MODE [IS] BLOCK [AT] <LINE [NUMBER] <<i>id-1</i> <i>integer-1</i>> <COLUMN COL> [NUMBER] <<i>id-2</i> <i>integer-2</i>>> [AT] <<i>id-3</i> <i>integer-3</i>> WITH <<BELL BEEP> BLINK HIGHLIGHT REVERSE-VIDEO UNDERLINE BLANK <SCREEN LINE> SIZE [IS] <<i>id-4</i> <i>integer-4</i>> FOREGROUND-COLOR [IS] <i>integer-5</i> BACKGROUND-COLOR [IS] <i>integer-6</i>>> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>		

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <id-1 literal-1> INTO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
DROP	<ul style="list-style-type: none"> Format DROP <id literal> FROM file-name 	V				
ENTER	<ul style="list-style-type: none"> Format ENTER language-name [routine-name] [.] 	V				

1-78 Supported COBOL Statements
Supported COBOL/400 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ENTRY	<ul style="list-style-type: none"> Format ENTRY <i>literal-1</i> [USING <i>id-1</i>] 	V	S	<i>id-1</i>		N
EVALUATE	<ul style="list-style-type: none"> Format EVALUATE <<i>id-1</i> <i>literal-1</i> <i>expr-1</i> TRUE FALSE> [ALSO <<i>id-2</i> <i>literal-2</i> <i>expr-2</i> TRUE FALSE>] WHEN <i>phrase-1</i> [ALSO <i>phrase-2</i>] <i>imperative-stmt-1</i> [WHEN OTHER <i>imperative-stmt-2</i>] [END-EVALUATE] Phrases <i>phrase-1</i> and <i>phrase-2</i> should be represented in the following form: <ANY <i>condition-1</i> TRUE FALSE [NOT] <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> [<THROUGH THRU> <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>>]> 	V	S	<i>id-1</i> <i>expr-1</i> <i>id-2</i> <i>expr-2</i> <i>condition-1</i> <i>id-3</i> <i>id-4</i> <i>arithmetic-expr1</i> <i>arithmetic-expr2</i>		N
EXIT	<ul style="list-style-type: none"> Format <i>paragraph-name</i>. EXIT. 	V	S			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
GOBACK	<ul style="list-style-type: none"> Format GOBACK 	V	S			N
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO [TO] <i>procedure-name-1</i> 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO [TO] <i>procedure-name-1</i> DEPENDING [ON] <i>id-1</i> 	V	S	<i>id-1</i>		N
	<ul style="list-style-type: none"> Format 3: altered <i>paragraph-name</i>. GO [TO]. 	V	S			N
	<ul style="list-style-type: none"> Format 4: MORE-LABELS GO [TO] MORE-LABELS 	N	N			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
IF	<ul style="list-style-type: none"> Format IF <i>condition-1</i> THEN <<i>stmt-1</i> NEXT SENTENCE> [ELSE <<i>stmt-2</i> NEXT SENTENCE>] [END-IF] 	V	S	<i>condition-1</i>		N
INITIALIZE	<ul style="list-style-type: none"> Format INITIALIZE <i>id-1</i> [REPLACING <ALPHABETIC ALPHANUMERIC NUMERIC ALPHANUMERIC-EDITED NUMERIC-EDITED> [DATA] BY <<i>id-2</i> <i>literal-1</i>>] 	V	S	<i>id-2</i>	<i>id-1</i>	N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-5</i> <i>id-4</i> <i>id-3</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 4 INSPECT <i>id-1</i> CONVERTING <<i>id-6</i> <i>literal-4</i>> TO <<i>id-7</i> <i>literal-5</i>> [BEFORE AFTER] [INITIAL] <<i>id-4</i> <i>literal-2</i>> 	V	S	<i>id-1</i> <i>id-6</i> <i>id-7</i> <i>id-4</i>	<i>id-1</i>	N
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> 	V	S	<i>file-name-1</i> <i>data-name-1</i>	<i>file-name-4</i>	N
	<ul style="list-style-type: none"> [[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>] 	VO	N	<i>file-name-2</i> <i>file-name-3</i>		N
	<ul style="list-style-type: none"> USING <i>file-name-2</i> <i>file-name-3</i> <OUTPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i>> 	V	S			N

1-80 Supported COBOL Statements
Supported COBOL/400 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <id-1 literal-1> TO id-2 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> id-1 TO id-2 	V	S	id-1	id-2	N
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <id-1 literal-1> BY id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1 id-2	id-3	S
OPEN	<ul style="list-style-type: none"> Format 1: sequential files OPEN <INPUT file-name-1 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
	[REVERSED [WITH] NO REWIND]	VO	N			
	OUTPUT file-name-2	V	S			
	[[WITH] NO REWIND]	VO	N			
	I-O file-name-3 EXTEND file-name-4>	V	S			
	<ul style="list-style-type: none"> Format 2: indexed and relative files OPEN <INPUT file-name-1 OUTPUT file-name-2 I-O file-name-3 EXTEND file-name-4> 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
<ul style="list-style-type: none"> Format 3: transaction OPEN I-O file-name 	V	S	file-name-1	file-name-1	N	

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM <procedure-name-1 [<THROUGH THRU> procedure-name-2] imperative-stmt-1 END-PERFORM> 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM <procedure-name-1 [<THROUGH THRU> procedure-name-2] <id-1 integer-1> TIMES <id-1 integer-1> TIMES imperative-stmt-1 END-PERFORM> 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	condition-1		N
	<ul style="list-style-type: none"> Format 4: with UNTIL phrase, without END-PERFORM PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] [[WITH] TEST <BEFORE AFTER>] UNTIL condition-1 	V	S	condition-1		N
	<ul style="list-style-type: none"> Format 5: with VARYING phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	id-3 id-4 condition-1	id-2	N
	<ul style="list-style-type: none"> Format 6: with VARYING phrase, without END-PERFORM PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 [AFTER <id-5 index-name-3> FROM <id-6 index-name-4 literal-3> BY <id-7 literal-4> UNTIL condition-2] <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-2 	N	N			N

1-82 Supported COBOL Statements
Supported COBOL/400 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
READ	<ul style="list-style-type: none"> Format 1: sequential retrieval READ <i>file-name-1</i> [NEXT FIRST LAST PRIOR] [RECORD] [[WITH] NO LOCK] [FORMAT [IS] <<i>id-1</i> <i>literal-1</i>>] 	V	S	<i>file-name-1</i>	<i>file-name-1 id-1</i>	N
	[NEXT FIRST LAST PRIOR] [RECORD]	VO	N			
	[[WITH] NO LOCK] [FORMAT [IS] < <i>id-1</i> <i>literal-1</i> >]					
	[INTO <i>id-2</i>] [[AT] END <i>imperative-stmt-1</i>] [NOT [AT] END <i>imperative-stmt-2</i>] [END-READ]	V	S			
	<ul style="list-style-type: none"> Format 2: random retrieval READ <i>file-name-1</i> [RECORD] [INTO <i>id-1</i>] [[WITH] NO LOCK] [KEY [IS] <i>data-name-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [INVALID [KEY] <i>imperative-stmt-3</i>] [NOT INVALID [KEY] <i>imperative-stmt-4</i>] [END-READ] 	V		<i>file-name-1 data-name-1</i>	<i>file-name-1 id-1</i>	N
<ul style="list-style-type: none"> Format 3: transaction (non-subfile) READ <i>file-name-1</i> [RECORD] [INTO <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [TERMINAL [IS] <<i>id-3</i> <i>literal-2</i>>] [<INDICATOR INDICATORS INDIC> <IS ARE> <i>id-4</i>] [NO DATA <i>imperative-stmt-1</i>] [INVALID [KEY] <i>imperative-stmt-2</i>] [NOT INVALID [KEY] <i>imperative-stmt-3</i>] [END-READ] 	V	S	<i>id-2 id-3 id-4</i>	<i>id-1</i>		
<ul style="list-style-type: none"> Format 4: transaction (subfile) READ <i>file-name-1</i> [[NEXT] MODIFIED] [RECORD] [INTO <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [TERMINAL [IS] <<i>id-3</i> <i>literal-2</i>>] [<INDICATOR INDICATORS INDIC> <IS ARE> <i>id-4</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [[AT] END <i>imperative-stmt-3</i>] [NOT [AT] END <i>imperative-stmt-4</i>] [END-READ] 	V	S	<i>id-2 id-3 id-4</i>	<i>id-1</i>		
RELEASE	<ul style="list-style-type: none"> Format RELEASE <i>record-name-1</i> [FROM <i>id-1</i>] 	V	S	<i>record-name-1 id-1</i>	<i>record-name-1</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
RETURN	<ul style="list-style-type: none"> Format RETURN <i>file-name-1</i> [RECORD] [INTO <i>id-1</i>] [[AT] END <i>imperative-stmt-1</i>] [NOT [AT] END <i>imperative-stmt-2</i>] [END-RETURN] 	V	S	<i>file-name-1</i>	<i>id-1</i>	N
REWRITE	<ul style="list-style-type: none"> Format 1: REWRITE <i>file-name-1</i> [FROM <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-REWRITE] 	V	S	<i>file-name-1</i> <i>id-1</i>	<i>file-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: transaction (subfile) REWRITE SUBFILE <i>file-name-1</i> [FROM <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [TERMINAL [IS] <<i>id-3</i> <i>literal-2</i>>] [<INDICATOR INDICATORS INDIC> <IS ARE> <i>id-4</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-REWRITE] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>		
ROLLBACK	<ul style="list-style-type: none"> Format ROLLBACK 	V	S			N
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [[AT] END <i>imperative-stmt-1</i>] WHEN <i>condition-1</i> <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>condition-1</i>	<i>id-2</i>	N
SEARCH	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [[AT] END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> [IS] EQUAL [TO] <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> [IS] EQUAL [TO] <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>data-name-1</i>		N

1-84 Supported COBOL Statements
Supported COBOL/400 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SET	<ul style="list-style-type: none"> Format 1: basic table handling SET <index-name-1 id-1> TO <index-name-2 id-2 integer-1> 	V	S	index-name-2 id-2	index-name-1 id-1	N
	<ul style="list-style-type: none"> Format 2: adjusting indexes SET index-name-3 <UP BY DOWN BY> <id-3 integer-2> 	V	S	index-name-3 id-3	index-name-3	N
	<ul style="list-style-type: none"> Format 3: external switches SET mnemonic-name-1 TO <ON OFF> 	V	S		mnemonic-name-1	N
	<ul style="list-style-type: none"> Format 4: condition-names SET condition-name-1 TO TRUE 	V	S		condition-variable	S
	<ul style="list-style-type: none"> Format 5: USAGE IS POINTER data items SET <id-4 ADDRESS OF id-5> TO <id-6 ADDRESS OF id-7 NULL NULLS> 	V	S	id-6 id-7	id-4 id-5	N
SORT	<ul style="list-style-type: none"> Format SORT file-name-1 [ON] <ASCENDING DESCENDING> [KEY] data-name-1 [[WITH] DUPLICATES [IN] [ORDER]] 	V	S			N
	[[COLLATING] SEQUENCE [IS] alphabet-name-1]	VO	N			N
	<USING file-name-2 INPUT PROCEDURE [IS] procedure-name-1 [<THROUGH THRU> procedure-name-2]> <GIVING file-name-3 OUTPUT PROCEDURE [IS] procedure-name-3 [<THROUGH THRU> procedure-name-4]>	V	S			N
START	<ul style="list-style-type: none"> Format START file-name-1 [KEY [IS] <EQUAL [TO] = LESS [THAN] < GREATER [THAN] > NOT LESS [THAN] NOT < NOT GREATER [THAN] NOT > LESS [THAN] OR EQUAL [TO] <= GREATER [THAN] OR EQUAL [TO] >= > <EXTERNALLY-DESCRIBED-KEY data-name-1>] [FORMAT [IS] <id-1 literal-1>] [FORMAT [IS] <id-2 literal-1>] [INVALID [KEY] imperative-stmt-1] [NOT INVALID [KEY] imperative-stmt-2] [END-START] 	V	S	data-name-1 id-1 id-2		
STOP	<ul style="list-style-type: none"> Format STOP <RUN literal> 	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
STRING	<ul style="list-style-type: none"> Format STRING <id-1 literal-1> DELIMITED [BY] <id-2 literal-2 SIZE> INTO id-3 [[WITH] POINTER id-4] [[ON] OVERFLOW imperative-stmt-1] [NOT [ON] OVERFLOW imperative-stmt-2] [END-STRING] 	V	S	id-1 id-2 id-3 id-4	id-3 id-4	S
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <id-1 literal-1> FROM id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <id-1 literal-1> FROM <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> id-1 FROM id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	N
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING id-1 [DELIMITED [BY] [ALL] <id-2 literal-1> [OR [ALL] <id-3 literal-2>]] INTO id-4 [DELIMITER [IN] id-5] [COUNT [IN] id-6] [[WITH] POINTER id-7] [TALLYING [IN] id-8] [[ON] OVERFLOW imperative-stmt-1] [NOT [ON] OVERFLOW imperative-stmt-2] [END-UNSTRING] 	V	S	id-1 id-2 id-3 id-4 id-7 id-8	id-4 id-5 id-6 id-7 id-8	S

1-86 Supported COBOL Statements
Supported COBOL/400 statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
WRITE	<ul style="list-style-type: none"> Format 1: sequential files (basic) WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] <<BEFORE AFTER> [ADVANCING] <<<i>id-1</i> <i>integer-1</i>> [LINE LINES] <i>mnemonic-name-1</i> PAGE>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [END-WRITE] 	VO	N			N
	<ul style="list-style-type: none"> Format 3: indexed and relative files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: format file WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [<INDICATOR INDICATORS INDIC> [IS ARE] <i>id-3</i>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 5: transactions WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [TERMINAL [IS] <<i>id-3</i> <i>literal-2</i>>] [STARTING [AT] [LINE] <<i>id-4</i> <i>literal-3</i>>] [<BEFORE AFTER> ROLLING [LINE LINES] <<i>id-5</i> <i>literal-4</i>> [THROUGH THRU] <<i>id-6</i> <i>literal-5</i>> <UP DOWN> <<i>id-7</i> <i>literal-6</i>> [LINE LINES]] [<INDICATOR INDICATORS INDIC> [IS ARE] <i>id-3</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 5: transactions WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [FORMAT [IS] <<i>id-2</i> <i>literal-1</i>>] [TERMINAL [IS] <<i>id-3</i> <i>literal-2</i>>] [<INDICATOR INDICATORS INDIC> [IS ARE] <i>id-3</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N

Compile-directing statements and directives

Statements and directives	Format	Parser	CE
BASIS	<ul style="list-style-type: none"> Format [sequence-number] BASIS <basis-name literal-1> 	VO	N
CBL (PROCESS)	<ul style="list-style-type: none"> Format <CBL PROCESS> [options-list] 	VO	N
*CONTROL (*CBL)	<ul style="list-style-type: none"> Format <*CONTROL *CBL> <SOURCE NOSOURCE LIST NOLIST MAP NOMAP> 	VO	N
COPY	<ul style="list-style-type: none"> Format COPY <text-name-1 literal-1> [<OF IN> library-name [file-name]] [SUPPRESS] [REPLACING operand-1 BY operand-2] 	V	S
	<ul style="list-style-type: none"> Format 2: DDS translate COPY <DD-format-name DD-ALL-FORMATS DDR-format-name DDR-ALL-FORMATS DDS-format-name DDS-ALL-FORMATS DDSR-format-name DDSR-ALL-FORMATS> [-I -O -I-O] [-INDICATOR -INDICATORS -INDIC] [<OF IN> library-name>] file-name [SUPPRESS] [REPLACING operand-1 BY operand-2] 	V	S
DELETE	<ul style="list-style-type: none"> Format [sequence-number] DELETE sequence-number-field 	VO	N
EJECT	<ul style="list-style-type: none"> Format EJECT [.] 	VO	N
ENTER	<ul style="list-style-type: none"> Format ENTER language-name-1 [routine-name-1]. 	VO	N
INSERT	<ul style="list-style-type: none"> Format [sequence-number] INSERT sequence-number-field 	VO	N
READY or RESET TRACE	<ul style="list-style-type: none"> Format <READY RESET> TRACE. 	VO	N

1-88 Supported COBOL Statements
Supported COBOL/400 statements

Statements and directives	Format	Parser	CE
REPLACE	<ul style="list-style-type: none"> Format 1 REPLACE ==pseudo-text-1== BY ==pseudo-text-2==. 	V	S
	<ul style="list-style-type: none"> Format 2 REPLACE OFF. 	V	S
SERVICE LABEL	<ul style="list-style-type: none"> Format SERVICE LABEL 	VO	N
SERVICE RELOAD	<ul style="list-style-type: none"> Format SERVICE RELOAD <i>id-1</i> 	VO	N
SKIP1/2/3	<ul style="list-style-type: none"> Format <SKIP1 SKIP2 SKIP3>. 	VO	N
TITLE	<ul style="list-style-type: none"> Format TITLE <i>literal</i>. 	VO	N
USE	<ul style="list-style-type: none"> Format 1: EXCEPTION ERROR declarative USE AFTER [STANDARD] <EXCEPTION ERROR> PROCEDURE [ON] <file-name-1 INPUT OUTPUT I-O EXTEND> 	VO	N
	<ul style="list-style-type: none"> Format 2: LABEL declarative USE [GLOBAL] AFTER [STANDARD] [BEGINNING ENDING] [FILE REEL UNIT] LABEL PROCEDURE [ON] <file-name-1 INPUT OUTPUT I-O EXTEND> 	VO	N
	<ul style="list-style-type: none"> Format 3: DEBUGGING declarative USE [FOR] DEBUGGING [ON] <procedure-name-1 ALL PROCEDURES file-name ALL [REFERENCES OF] id-1> 	VO	N

Supported Unisys ASCII COBOL statements

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S
NULL / NULLS	V	S

Special registers

Constants	Parser	CE
DEBUG ITEM	V	N
LINAGE-COUNTER	V	N
LINE-COUNTER	V	N
PAGE-COUNTER	V	N

1-90 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Literals

Literal type	Format	Parser	CE
Nonnumeric literal	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
Numeric literals	<ul style="list-style-type: none"> Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format <i>text-name-1</i> [<IN OF> <i>library-name-1</i>] 	V	S
to Procedure Division	<ul style="list-style-type: none"> Format <i>paragraph-name-1</i> [<IN OF> <i>section-name-1</i>] 	V	S
to Data Division	<ul style="list-style-type: none"> Format 1: subscripting <i>data-name-1</i> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) (<i>leftmost-character-position: [length]</i>) 	V	S
	<ul style="list-style-type: none"> Format 2 <<i>condition-name-1</i> <i>data-name-1</i>> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 3 LINAGE-COUNTER [<IN OF> <i>file-name-2</i>] 	V	S
	<ul style="list-style-type: none"> Format 4 <PAGE-COUNTER LINE-COUNTER> <IN OF> <i>report-name</i> 	V	S

Division	Format	Parser	CE
Condition names	<ul style="list-style-type: none"> Format: Data Division <i>condition-name-1</i> [<IN OF> <i>data-name-1</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) 	V	S
Subscript	<ul style="list-style-type: none"> Format <<i>integer-1</i> ALL <i>data-name-3</i> [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>> 	V	S
Reference modification	<ul style="list-style-type: none"> Format <<i>data-name-1</i>> (<i>leftmost-character-position</i>: [<i>length</i>]) 	V	S

Control Division

Names	Format	Parser	CE
Control Division Entry	<ul style="list-style-type: none"> Format [CONTROL DIVISION]. [ALPHABET SECTION]. [SOURCE-ALPHABET CHARACTERS ARE <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]] 	N	N
Inline Section	<ul style="list-style-type: none"> Format APPLY INLINE <ADD ALL CONVERSION IF INSPECT MOVE SIGN> FOR <ALL PROCEDURES <i>procedure-name-1</i>> [<THROUGH THRU> <i>procedure-name-2</i>] 	N	N

Identification Division

Names	Format	Parser	CE
Identification Division	<ul style="list-style-type: none"> Format IDENTIFICATION DIVISION. PROGRAM-ID <i>program-name</i> . [AUTHOR. <i>[comment-entry]</i>] [INSTALLATION. <i>[comment-entry]</i>] [DATE-WRITTEN. <i>[comment-entry]</i>] [DATE-COMPILED. <i>[comment-entry]</i>] [SECURITY. <i>[comment-entry]</i>] [REMARKS. <i>[comment-entry]</i>]	V	S
	[REMARKS. <i>[comment-entry]</i>]	VO	N

Environment Division

Input-output section

Paragraphs and clauses	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential files FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i> [; RESERVE <i>integer</i> [AREA AREAS]] [; ORGANIZATION IS SEQUENTIAL] [; ACCESS MODE IS SEQUENTIAL] [FILE STATUS IS <i>data-name-1</i>].	V	S
	[; RESERVE <i>integer</i> [AREA AREAS]] [; ORGANIZATION IS SEQUENTIAL] [; ACCESS MODE IS SEQUENTIAL] [FILE STATUS IS <i>data-name-1</i>].	VO	N
	<ul style="list-style-type: none"> Format 2: indexed files FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i> [; RESERVE <i>integer</i> [AREA AREAS]] [; ORGANIZATION IS INDEXED] [; ACCESS MODE IS <SEQUENTIAL RANDOM DYNAMIC>] [; RECORD KEY IS <i>data-name-2</i> [; ALTERNATE RECORD KEY IS <i>data-name-3</i> [WITH DUPLICATES] [; FILE STATUS IS <i>data-name-1</i>].	V	S
	[; RESERVE <i>integer</i> [AREA AREAS]] [; ORGANIZATION IS INDEXED] [; ACCESS MODE IS <SEQUENTIAL RANDOM DYNAMIC>] [; RECORD KEY IS <i>data-name-2</i> [; ALTERNATE RECORD KEY IS <i>data-name-3</i> [WITH DUPLICATES] [; FILE STATUS IS <i>data-name-1</i>].	VO	N

	<ul style="list-style-type: none"> • Format 3: relative files FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i> 	V	S
	<pre>[; RESERVE <i>integer</i> [AREA AREAS]] ; ORGANIZATION IS RELATIVE [; ACCESS MODE IS <SEQUENTIAL [, RELATIVE KEY IS <i>data-name-1</i>] <RANDOM DYNAMIC> RELATIVE KEY IS <i>data-name-1</i>>] [; FILE STATUS IS <i>data-name-2</i>].</pre>	VO	S
	<ul style="list-style-type: none"> • Format 4: sort-merge FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i>. 	V	S
FILE-CONTROL paragraph	<ul style="list-style-type: none"> • Format 5: IBM tape processing FILE-CONTROL. SELECT [OPTIONAL] <i>file-name</i> ASSIGN TO <i>assignment-name</i> 	V	S
	<pre>[; RESERVE <<i>integer</i> NO> ALTERNATE [AREA AREAS]] [; ACCESS MODE IS SEQUENTIAL] [; ORGANIZATION IS SEQUENTIAL].</pre>	VO	N
I-O-CONTROL paragraph	<ul style="list-style-type: none"> • Format 1: sequential files I-O CONTROL. [; RERUN ON <<i>assignment-name-1</i> <i>file-name-1</i>> EVERY <<<i>integer-1</i> RECORDS END OF <REEL UNIT>> OF <i>file-name-1</i> <i>integer-2</i> CLOCK-UNITS <i>condition-name</i>>] [; SAME [RECORD] AREA FOR <i>file-name-3</i>] [; MULTIPLE FILE TAPE CONTAINS <i>file-name-5</i> [POSITION <i>integer-3</i>]]. 	VO	N
	<ul style="list-style-type: none"> • Format 2: indexed and relative files I-O CONTROL. [RERUN ON <<i>assignment-name-1</i> <i>file-name-1</i>> EVERY <<i>integer-1</i> RECORDS OF <i>file-name-1</i> <i>integer-2</i> CLOCK-UNITS <i>condition-name</i>>] [; SAME [RECORD] AREA FOR <i>file-name-3</i>]. 	VO	N
	<ul style="list-style-type: none"> • Format 3: sort-merge I-O CONTROL. [; SAME <RECORD SORT SORT-MERGE> AREA FOR <i>file-name-1</i>]. 	VO	N
	<ul style="list-style-type: none"> • Format 4: IBM tape processing I-O CONTROL. [; RERUN ON <i>assignment-name-1</i> EVERY <<i>integer-1</i> RECORDS [END OF] REEL> OF <i>file-name-1</i>] [; SAME [RECORD] AREA FOR <i>file-name-2</i>] 	VO	N

1-94 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

	<ul style="list-style-type: none"> • Format 5: direct I-O I-O CONTROL. [; APPLY INTERLOCK ON <i>file-name-1</i>] [; APPLY PROTECT ON <i>file-name-2</i> [FOR <i>integer-3</i> RECORDS]] 	VO	N
	<ul style="list-style-type: none"> • Format 6: indexed sequential I-O I-O CONTROL. [; APPLY INTERLOCK ON <i>file-name-1</i>] [; APPLY PROTECT ON <i>file-name-2</i> [FOR <i>integer-3</i> RECORDS]] [; APPLY CORE-INDEX ON <i>file-name-3</i>] 	VO	N

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COM-PUTER paragraph	<ul style="list-style-type: none"> • Format OBJECT-COMPUTER. computer-name [, MEMORY SIZE <i>integer</i> <WORDS CHARACTERS MODULES>] [, PROGRAM COLLATING SEQUENCE IS <i>alphabet-name</i>]. 	VO	N
SOURCE-COM-PUTER paragraph	<ul style="list-style-type: none"> • Format SOURCE-COMPUTER. computer-name. 	VO	N
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> • Format SPECIAL-NAMES. [, <i>implementor-name</i> <IS <i>mnemonic-name</i> [, ON STATUS IS <i>condition-name-1</i>, [OFF STATUS IS <i>condition-name-2</i>]] IS <i>mnemonic-name</i> [, OFF STATUS IS <i>condition-name-2</i>, [ON STATUS IS <i>condition-name-1</i>]] ON STATUS IS <i>condition-name-1</i>, [OFF STATUS IS <i>condition-name-2</i>] OFF STATUS IS <i>condition-name-2</i>, [ON STATUS IS <i>condition-name-1</i>]->] 	V	PS
	[, <i>alphabet-name</i> IS <STANDARD-1 NATIVE <i>implementor-name</i> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i>]->]	VO	N
	[, CURRENCY SIGN IS <i>literal-4</i>]	VO	N
	[, DECIMAL POINT IS COMMA]	V	S
	[, CLASS NAME. IS [, <i>mnemonic-name</i>] [VALUE IS <i>literal-5</i> [<THROUGH THRU> <i>literal-6</i>]]]	VO	N
ACTUAL KEY clause	<ul style="list-style-type: none"> • Format ACTUAL KEY IS <i>data-name</i> 	VO	N

APPLY clause	<ul style="list-style-type: none"> Format APPLY <EXREF EXDEF> ON <i>id-1</i> 	VO	N
APPLY WRITE-ONLY clause	<ul style="list-style-type: none"> Format APPLY WRITE-ONLY ON <i>file-name-1</i> 	VO	N
ASSIGN clause	<ul style="list-style-type: none"> Format ASSIGN TO <i>assignment-name</i> 	V	S
FILE-LIMITS clause	<ul style="list-style-type: none"> Format [: <FILE-LIMIT IS FILE-LIMITS ARE> <i><data-name-1 literal-1></i> <THROUGH THRU> <i><data-name-2 literal-2></i>] [: <i><data-name-3 literal-3></i> <THROUGH THRU> <i><data-name-4 literal-4></i>] 	VO	N
PROCESSING MODE clause	<ul style="list-style-type: none"> Format PROCESSING MODE IS SEQUENTIAL 	VO	N
RERUN clause	<ul style="list-style-type: none"> Format RERUN [ON <i><file-name-1 UNISERVO UNISERVOS MASS-STORAGE></i>] EVERY <<<i>integer-1</i> RECORDS END OF <REEL UNIT>> OF <i>file-name-1</i> 	VO	N
RESERVE clause	<ul style="list-style-type: none"> Format RESERVE <<i>integer-1</i> NO> ALTERNATE [AREA AREAS] 	VO	N
SEGMENT-LIMIT clause	<ul style="list-style-type: none"> Format SEGMENT LIMIT IS <i>segment-number</i> 	VO	N
WITH DEBUGGING MODE clause	<ul style="list-style-type: none"> Format WITH DEBUGGING MODE 	VO	N
WITH RANK clause	<ul style="list-style-type: none"> Format WITH RANK OF <i>integer</i> 	VO	N

Data Division

Entries and Clauses	Format	Parser	CE
Communication Description Entry	<ul style="list-style-type: none"> Format 1 CD <i>cd-name</i>; FOR [INITIAL] INPUT [; SYMBOLIC QUEUE IS <i>data-name-1</i>] [; SYMBOLIC SUB-QUEUE-1 IS <i>data-name-2</i>] [; SYMBOLIC SUB-QUEUE-2 IS <i>data-name-3</i>] [; SYMBOLIC SUB-QUEUE-3 IS <i>data-name-4</i>] [; MESSAGE DATE IS <i>data-name-5</i>] [; MESSAGE TIME IS <i>data-name-6</i>] [; SYMBOLIC SOURCE IS <i>data-name-7</i>] [; TEXT LENGTH IS <i>data-name-8</i>] [; END KEY IS <i>data-name-9</i>] [; STATUS KEY IS <i>data-name-10</i>] [; MESSAGE COUNT IS <i>data-name-11</i>] 	N	N
	<ul style="list-style-type: none"> Format 2 CD <i>cd-name</i>; FOR OUTPUT [; DESTINATION COUNT IS <i>data-name-1</i>] [; TEXT LENGTH IS <i>data-name-2</i>] [; STATUS KEY IS <i>data-name-3</i>] [; DESTINATION TABLE OCCURS <i>integer-2</i> TIMES [; INDEXED BY <i>index-name-1</i>]] [; ERROR KEY IS <i>data-name-4</i>] [; SYMBOLIC DESTINATION IS <i>data-name-5</i>] 	N	N
Data Description Entry	<ul style="list-style-type: none"> Format 1 <i>level-number</i> <<i>data-name-1</i> FILLER> [; REDEFINES <i>data-name-2</i>] [; <PICTURE PIC> IS <i>character-string</i>] [[USAGE IS] <COMPUTATIONAL COMP DISPLAY COMPUTATIONAL-1 COMP-1 COMPUTATIONAL-2 COMP-2 DISP>] [; [SIGN IS] <LEADING TRAILING> [SEPARATE CHARACTER]] [; <SYNCHRONIZED SYNC> [LEFT RIGHT]] [; <JUSTIFIED JUST> RIGHT] [; BLANK WHEN ZERO] [; VALUE IS <i>literal</i>]. 	V	S
	<ul style="list-style-type: none"> Format 2 66 <i>data-name-1</i>; RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>]. 	V	S
	<ul style="list-style-type: none"> Format 3 88 <i>condition-name</i>; <VALUE IS VALUES ARE> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>] [, <i>literal-3</i> [<THROUGH THRU> <i>literal-4</i>]]. 	V	S

Entries and Clauses	Format	Parser	CE
File Description (FD) Entry	<ul style="list-style-type: none"> Format 1: sequential files <p><i>FD file-name</i></p>	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; DATA <RECORD IS RECORDS ARE> <i>data-name-2</i>] [; LINAGE IS <<i>data-name-3</i> <i>integer-5</i>> LINES [, WITH FOOTING AT <<i>data-name-4</i> <i>integer-6</i>>] [, LINES AT TOP <<i>data-name-6</i> <i>integer-7</i>>] [, LINES AT BOTTOM <<i>data-name-7</i> <i>integer-8</i>>]] [; CODE-SET IS <i>alphabet-name</i>] DENSITY IS <6 8 12> LINES.</pre>	VO	N
	<ul style="list-style-type: none"> Format 2: indexed and relative files <p><i>FD file-name</i></p>	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-2</i>].</pre>	VO	N
	<ul style="list-style-type: none"> Format 3: IBM tape processing <p><i>FD file-name</i></p>	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; DATA <RECORD [IS] RECSORDS [ARE]> <i>data-name-2</i>]. ; RECORDING MODE IS <E U V> [AN]</pre>	VO	N
	<ul style="list-style-type: none"> Format 4: sort-merge <p><i>SD file-name</i></p>	V	S
	<pre>[; RECORD CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> CHARACTERS] [; DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-1</i>].</pre>	VO	N

1-98 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Entries and Clauses	Format	Parser	CE
	<ul style="list-style-type: none"> Format 5: report writer FD <i>file-name</i> 	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; CODE-SET IS <i>alphabet-name</i>] ; <REPORT IS REPORTS ARE> <i>report-name</i> DENSITY IS <6 8 12> LINES.</pre>	VO	N
Report Description (RD) Entry	<ul style="list-style-type: none"> Format RD <i>report-name</i> [; CODE <i>literal-1</i>] [; <CONTROL IS CONTROLS ARE> <<i>data-name-1</i> FINAL [, <i>data-name-1</i>]>] [; PAGE [LIMIT IS LIMITS ARE] <i>integer-1</i> [LINE LINES] [, HEADING <i>integer-2</i>] [, FIRST DETAIL <i>integer-3</i>] [, LAST DETAIL <i>integer-4</i>] [, FOOTING <i>integer-5</i>]. 	V	S
Report Group Description	<ul style="list-style-type: none"> Format 1 01 [<i>data-name-1</i>] [; LINE NUMBER IS <<i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i>>] [; NEXT GROUP IS <<i>integer-3</i> PLUS <i>integer-4</i> NEXT PAGE>] [; TYPE IS <REPORT HEADING RH PAGE HEADING PH <CONTROL HEADING CH> <<i>data-name-2</i> FINAL> DETAIL DE <CONTROL FOOTING CF> <<i>data-name-3</i> FINAL> PAGE FOOTING PF REPORT FOOTING RF> [; [USAGE IS] <DISPLAY DISP>] Format 2 <i>level-number</i> [<i>data-name-1</i>] [; LINE NUMBER IS <<i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i>>] [; [USAGE IS] <DISPLAY DISP>] Format 3 <i>level-number</i> [<i>data-name-1</i>] [; BLANK WHEN ZERO] [; GROUP INDICATE] [; <JUSTIFIED JUST> RIGHT] [; LINE NUMBER IS <<i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i>>] [; COLUMN NUMBER IS <i>integer-3</i>] [; <PICTURE PIC> IS <i>character-string</i> ; SOURCE IS <i>id-1</i> < ; VALUE IS <i>literal</i> ; SUM <i>id-2</i> [UPON <i>data-name-2</i>]> [; RESET ON <<i>data-name-3</i> FINAL>] [; [USAGE IS] <DISPLAY DISP>] 	V	S
Saved-Area Description	<ul style="list-style-type: none"> Format SA <i>area-name</i> ; AREA CONTAINS <i>integer-1</i> RECORDS [; WITH RANK OF <i>integer-2</i>] 	N	N

Entries and Clauses	Format	Parser	CE
AREA CONTAINS clause	<ul style="list-style-type: none"> Format AREA CONTAINS <i>integer-1</i> RECORDS 	N	N
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> Format BLANK WHEN ZERO 	V	S
BLOCK CONTAINS clause	<ul style="list-style-type: none"> Format BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS> 	N	N
CODE clause	<ul style="list-style-type: none"> Format CODE <i>literal-1</i> 	V	N
CODE-SET clause	<ul style="list-style-type: none"> Format CODE-SET IS <i>alphabet-name</i> 	V	N
COLUMN NUMBER clause	<ul style="list-style-type: none"> Format COLUMN NUMBER IS <i>integer-1</i> 	V	N
CONTROL clause	<ul style="list-style-type: none"> Format <CONTROL IS CONTROLS ARE> <i><data-name-1 FINAL></i> 	V	N
Data-name or FILLER clause	<ul style="list-style-type: none"> Format <i><data-name FILLER></i> 	V	S
DATA RECORDS clause	<ul style="list-style-type: none"> Format DATA <RECORD IS RECORDS ARE> <i>data-name</i> 	V	S
DENSITY clause	<ul style="list-style-type: none"> Format DENSITY IS <6 8 12> LINES 	VO	N
JUSTIFIED clause	<ul style="list-style-type: none"> Format <JUSTIFIED JUST> RIGHT 	V	S
LABEL RECORDS clause	<ul style="list-style-type: none"> Format LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> 	V	S

1-100 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Entries and Clauses	Format	Parser	CE
LINAGE clause	<ul style="list-style-type: none"> Format LINAGE IS <data-name-1 integer-1> LINES [, WITH FOOTING AT <data-name-2 integer-2>] [, LINES AT TOP <data-name-3 integer-3>] [, LINES AT BOTTOM <data-name-4 integer-4>] 	N	N
LINE NUMBER clause	<ul style="list-style-type: none"> Format LINE NUMBER IS <integer-1 [ON NEXT PAGE] PLUS integer-2> 	V	N
NEXT GROUP clause	<ul style="list-style-type: none"> Format NEXT GROUP IS <integer-1 PLUS integer-2 NEXT PAGE> 	V	N
OCCURS clause	<ul style="list-style-type: none"> Format 1: fixed-length tables OCCURS integer-2 TIMES [<ASCENDING DESCENDING> KEY IS data-name-2] [[INDEXED BY index-name-1] 	V	S
	<ul style="list-style-type: none"> Format 2: variable-length tables OCCURS integer-1 TO integer-2 TIMES DEPENDING ON data-name-1 [<ASCENDING DESCENDING> KEY IS data-name-2] [[INDEXED BY index-name-1] 	V	S
PAGE clause	<ul style="list-style-type: none"> Format PAGE [LIMIT IS LIMITS ARE] integer-1 [LINE LINES] [, HEADING integer-2] [, FIRST DETAIL integer-3] [, LAST DETAIL integer-4] [, FOOTING integer-5] 	V	N
PICTURE clause	<ul style="list-style-type: none"> Format <PICTURE PIC> IS character-string 	V	S
POINT LOCATION clause	<ul style="list-style-type: none"> Format POINT LOCATION IS <LEFT RIGHT> integer-1 <PLACE PLACES> 	N	N
RECORD CONTAINS clause	<ul style="list-style-type: none"> Format RECORD CONTAINS integer-1 TO integer-2 CHARACTERS 	VO	N

Entries and Clauses	Format	Parser	CE
RECORDING MODE clause	<ul style="list-style-type: none"> Format 1: IBM tape processing RECORDING MODE IS <E U V> [AN] 	V	N
	<ul style="list-style-type: none"> Format 2: sequential I-O RECORDING MODE IS <<LION CFH> [AN] <FORMO1 FORMO2 FORMO3> [U] BLANK [SIGN] COMPACT INTERNAL SIGN SDF> 	V	N
	<ul style="list-style-type: none"> Format 3: direct I-O RECORDING MODE IS <BLANK [SIGN] COMPACT INTERNAL SIGN SDF> 	V	N
	<ul style="list-style-type: none"> Format 4: Information Interchange tape processing RECORDING MODE IS <D F S U> 	V	N
REDEFINES clause	<ul style="list-style-type: none"> Format <i>level-number data-name-1; REDEFINES data-name-2</i> 	V	S
RENAMES clause	<ul style="list-style-type: none"> Format <i>66 data-name-1; RENAMES data-name-2 [<THROUGH THRU> data-name-3].</i> 	V	S
REPORT clause	<ul style="list-style-type: none"> Format <REPORT IS REPORTS ARE> <i>report-name</i> 	VO	N
SIGN clause	<ul style="list-style-type: none"> Format [SIGN IS] <LEADING TRAILING> [SEPARATE CHARACTER] 	V	S
SOURCE clause	<ul style="list-style-type: none"> Format SOURCE IS <i>id-1</i> 	V	N
SUM clause	<ul style="list-style-type: none"> Format SUM <i>id-1</i> [UPON <i>data-name-2</i>] [RESET ON <<i>data-name-3</i> FINAL>] 	V	N
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZE SYNC> [LEFT RIGHT] 	V	N
TYPE clause	<ul style="list-style-type: none"> Format TYPE IS <REPORT HEADING RH PAGE HEADING PH <CONTROL HEADING CH> <<i>data-name-2</i> FINAL> DETAIL DE <CONTROL FOOTING CF> <<i>data-name-3</i> FINAL> PAGE FOOTING PF REPORT FOOTING RF> 	V	N

1-102 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Entries and Clauses	Format	Parser	CE
USAGE clause	<ul style="list-style-type: none"> • Format 1: general [USAGE IS] <COMPUTATIONAL COMP DISPLAY COMPUTATIONAL-1 COMP-1 COMPUTATIONAL-2 COMP-2 DISP> 	V	S
	<ul style="list-style-type: none"> • Format 2: table handling [USAGE IS] INDEX 	V	S
	<ul style="list-style-type: none"> • Format 3: asynchronous processing [USAGE IS] LOCK 	N	N
VALUE clause	<ul style="list-style-type: none"> • Format 1 VALUE IS <i>literal</i> 	V	S
	<ul style="list-style-type: none"> • Format 2 <VALUE IS VALUES ARE> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>] [, <i>literal-3</i> [<THROUGH THRU> <i>literal-4</i>]] 	V	S
VALUE OF clause	<ul style="list-style-type: none"> • Format 1 VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>> 	VO	N
	<ul style="list-style-type: none"> • Format 2: sequential and direct I-O VALUE OF <<i>system-name-1</i> FILE-ID SET-ID CREATION-DATE PURGE-DATE FILE-ACCESS FILE-QUALIFIER> IS <<i>data-name-1</i> <i>literal-1</i>> 	VO	N

Procedure Division

Arithmetic operators

Binary operation	Meaning	Format	Parser	CE			
				Gen.	Read variables	Written variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

1-104 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Conditional Expressions

Conditions	Format	Parser	CE			
			Gen.	Read variables	Written variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id</i> IS [NOT] <NUMERIC ALPHABETIC CLASS-NAME <i>mnemonic-name</i>> 	V	S	<i>id-1</i>		S
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format <i>oper-1</i> IS [NOT] <GREATER THAN LESS THAN EQUAL TO> <i>oper-2</i> <i>oper-1</i>: <<i>id-1</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>oper-2</i>: <i>id-2</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> IS [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	<i>operand-1</i>		S
Switch-status condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT <i>simple-condition</i> 	V	S	<i>condition-1</i>		S
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1</i> <AND OR> <i>condition-2</i> 	V	S	<i>condition-1</i> <i>condition-2</i>		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format <i>relation-condition</i> <AND OR> [NOT] [<i>relational-operator</i>] <i>object</i> 	V	S	<i>relation-condition</i> <i>object</i>		S

Statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format PROCEDURE DIVISION [USING <i>data-name-1</i>] [WITH ENTRY POINTS <i>procedure-name-1</i>] 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N
DECLARATIVES	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FROM <i>mnemonic-name</i>] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE [YYYYMMDD] DAY [YYYYDDD] TIME DATE-TIME [MMDYYYY]> 	VO	N			N
	<ul style="list-style-type: none"> Format 3: ACCEPT <i>id</i> [FROM <CARD-READER CONSOLE DATE-TIME>] 	VO	N			N
ACCEPT MESSAGE COUNT	<ul style="list-style-type: none"> Format ACCEPT <i>cd-name</i> MESSAGE COUNT 	N	N			N
ADD	<ul style="list-style-type: none"> Format 1 ADD <<i>id-1</i> <i>literal-1</i>> TO <i>id-2</i> [ROUNDED] [: ON SIZE ERROR <i>imperative-stmf</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <<i>id-1</i> <i>literal-1</i>>, <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [:ON SIZE ERROR <i>imperative-stmf</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
ADD	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> [ROUNDED] [:ON SIZE ERROR <i>imperative-stmf</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N

1-106 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
ALL	<ul style="list-style-type: none"> Format ALL <INACTIVE ACTIVE> FOR <area-name-1 section-name-1> 	N	N			N
ALTER	<ul style="list-style-type: none"> Format ALTER <i>procedure-name-1</i> TO [PROCEED TO] <i>procedure-name-2</i> 	VO	N			N
ANY	<ul style="list-style-type: none"> Format ANY <INACTIVE ACTIVE> FOR <area-name-1 section-name-1> 	N	N			N
CALL	<ul style="list-style-type: none"> Format 1: general CALL <<i>id-1</i> <i>literal-1</i>> [USING <<i>data-name-1</i> <i>literal-2</i>>] [; ON OVERFLOW <i>imperative-stmt</i>] 	V	S	<i>id-1</i> <i>data-name-1</i>		S
	<ul style="list-style-type: none"> Format 2: CALL debug CALL 'PAD\$\$' 	V	S			N
CANCEL	<ul style="list-style-type: none"> Format CANCEL <<i>id-1</i> <i>literal-1</i>> 	V	S			N
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> [<REEL UNIT> [FOR REMOVAL WITH NO REWIND] WITH <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files CLOSE <i>file-name-1</i> [WITH LOCK] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 3: IBM tape processing CLOSE <i>file-name-1</i> [<REEL UNIT> [WITH NO REWIND] WITH <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	VO	N			N	

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <i>id-1</i> [ROUNDED] = <i>arithmetic-expr</i> [; ON SIZE ERROR <i>imperative-stmt</i>] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S
COPY	<ul style="list-style-type: none"> Format COPY <i>text-name</i> [<OF IN> <i>library-name</i>] [REPLACING, <==<i>pseudo-text-1</i>== <i>id-1</i> <i>literal-1</i> <i>word-1</i>> BY <==<i>pseudo-text-2</i>== <i>id-2</i> <i>literal-2</i> <i>word-2</i>>] 	V	S			N
DELETE	<ul style="list-style-type: none"> Format: indexed and relative files DELETE <i>file-name</i> RECORD [; INVALID KEY <i>imperative-stmt</i>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISABLE	<ul style="list-style-type: none"> Format DISABLE <INPUT [TERMINAL] OUTPUT> <i>cd-name</i> WITH KEY <<i>id-1</i> <i>literal-1</i>> 	N	N			N
DISPLAY	<ul style="list-style-type: none"> Format 1 DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <CARD-PUNCH CONSOLE PRINTER>] 	V	S	<i>id-1</i>		S
		VO	N			N
	<ul style="list-style-type: none"> Format 2 DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <i>mnemonic-name</i>] 	V	S	<i>id-1</i>		S

1-108 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <id-1 literal-1> INTO id-2 [ROUNDED] [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3 id-4	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3 id-4	S
ENABLE	<ul style="list-style-type: none"> Format ENABLE <INPUT [TERMINAL] OUTPUT> cd-name WITH KEY <id-1 literal-1> 	V	S	cd-name id-1		N
ENTER	<ul style="list-style-type: none"> Format 1 ENTER <MASM ASM PL1 FTN> literal [USING argument] 	V	S	argument	argument	N
	<ul style="list-style-type: none"> Format 2 ENTER MASM "C\$INFO" USING file-name, data-name 	V	S	file-name data-name	file-name data-name	N
	<ul style="list-style-type: none"> Format 3 ENTER <FORTRAN FD [ASM]> literal [USING argument] 	V	S	argument	argument	N

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
EXAMINE	<ul style="list-style-type: none"> Format EXAMINE <i>id-1</i> <TALLYING <UNTIL FIRST ALL LEADING TRAILING> <<i>id-2</i> <i>literal-1</i>> [REPLACING BY <<i>id-3</i> <i>literal-2</i>>] REPLACING <[UNTIL] FIRST ALL LEADING TRAILING> <<i>id-2</i> <i>literal-1</i>> BY <<i>id-3</i> <i>literal-2</i>>> 	V	S	<i>id-1</i>		N
EXHIBIT	<ul style="list-style-type: none"> Format EXHIBIT <NAMED CHANGED CHANGED NAMED> <<i>data-name-1</i> <i>literal-1</i>> 	V	S	<i>data-name-1</i>		N
EXIT	<ul style="list-style-type: none"> Format <i>paragraph-name</i>. EXIT. 	V	S			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
FREE	<ul style="list-style-type: none"> Format FREE <i>file-name</i> RECORD 	V	S			N
GENE-RATE	<ul style="list-style-type: none"> Format GENERATE <<i>data-name</i> <i>report-name</i>> 	V	S	<i>data-name</i>		N
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO TO <i>procedure-name-1</i> 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO TO <i>procedure-name-1</i> DEPENDING ON <i>id-1</i> 	V	S	<i>id-1</i>		N
HOLD	<ul style="list-style-type: none"> Format HOLD <ALL <i>are-name-1</i> <i>section-name-1</i>> 	N	N			N
IF	<ul style="list-style-type: none"> Format IF <i>condition</i>; <<i>stmt-1</i> NEXT SENTENCE>; ELSE <<i>stmt-2</i> NEXT SENTENCE> 	V	S	<i>condition-1</i>		N
INITIATE	<ul style="list-style-type: none"> Format INITIATE <i>report-name-1</i> 	V	S		<i>report-name-1</i>	N

1-110 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING , <i>id-2</i> FOR, <CHARACTERS <TRAILING ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] , <TRAILING ALL LEADING FIRST> , <<i>id-5</i> <i>literal-3</i>> BY <<i>id-7</i> <i>literal-5</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-5</i> <i>id-6</i> <i>id-7</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING , <i>id-2</i> FOR, <CHARACTERS <TRAILING ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] , <TRAILING ALL LEADING FIRST> , <<i>id-5</i> <i>literal-3</i>> BY <<i>id-7</i> <i>literal-5</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i> <i>id-7</i>	<i>id-1</i>	N
LOCK	<ul style="list-style-type: none"> Format LOCK <i>id-1</i> 	N	N			N
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> ON <ASCENDING DESCENDING> KEY <i>data-name-1</i> [COLLATING SEQUENCE IS <i>alphabet-name-1</i>] 	V	S	<i>file-name-1</i> <i>data-name-1</i>	<i>file-name-4</i>	N
	<ul style="list-style-type: none"> USING <i>file-name-2</i> <i>file-name-3</i> <OUTPUT PROCEDURE IS <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i>> 	VO	N	<i>file-name-2</i> <i>file-name-3</i>		N
	<ul style="list-style-type: none"> Format MONITOR <<i>procedure-name-1</i> <i>id-1</i> ALL> [UNTIL <i>procedure-name</i>] 	V	S			N
MONITOR	<ul style="list-style-type: none"> Format MONITOR <<i>procedure-name-1</i> <i>id-1</i> ALL> [UNTIL <i>procedure-name</i>] 	VO	N			N

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <id-1 literal-1> TO id-2 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> id-1 TO id-2 	V	S	id-1	id-2	N
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <id-1 literal-1> BY id-2 [ROUNDED] [: ON SIZE ERROR imperative-stmt] 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [: ON SIZE ERROR imperative-stmt] 	V	S	id-1 id-2	id-3	S
NOTE	<ul style="list-style-type: none"> Format NOTE character-string. 	VO	N			N
ON	<ul style="list-style-type: none"> Format ON integer-1 [AND EVERY integer-2] [UNTIL integer-3] ; <imperative-stmt-1 NEXT SENTENCE> [: <OTHERWISE ELSE <imperative-stmt-2 NEXT SENTENCE>] 	V	S			N
OPEN	<ul style="list-style-type: none"> Format 1: sequential files OPEN <INPUT file-name-1 [REVERSED WITH NO REWIND] OUTPUT file-name-2 [WITH NO REWIND] I-O file-name-3 EXTEND file-name-4> 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
		VO	N			N
		V	S			N
		VO	N			N
		V	S			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files OPEN <INPUT file-name-1 OUTPUT file-name-2 I-O file-name-3> 	V	S			file-name-1 file-name-2 file-name-3

1-112 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <i><id-1 integer-1></i> TIMES 	V	S	<i>id-1</i>		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] UNTIL <i>condition-1</i> 	V	S	<i>condition-1</i>		N
	<ul style="list-style-type: none"> Format 4: with VARYING phrase PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] VARYING <i><id-2 index-name-1></i> FROM <i><id-3 index-name-2 literal-1></i> BY <i><id-4 literal-2></i> UNTIL <i>condition-1</i> [AFTER <i><id-5 index-name-3></i> FROM <i><id-6 index-name-4 literal-3></i> BY <i><id-7 literal-4></i> UNTIL <i>condition-2</i>] 	V	S	<i>id-3</i> <i>id-4</i> <i>condition-1</i> <i>id-6</i> <i>id-7</i> <i>condition-2</i>	<i>id-2</i> <i>id-5</i>	N
PROCESS	<ul style="list-style-type: none"> Format PROCESS <i>section-name</i> [FROM <i>id</i> [USING <i>area-name</i>]] 	N	N			N
READ	<ul style="list-style-type: none"> Format 1: sequential retrieval READ <i>file-name-1</i> 	V	S	<i>file-rec-1</i> (corresponding to <i>file-name-1</i>)	<i>file-rec-1</i> (corresponding to <i>file-name-1</i>) <i>id-1</i>	N
	[NEXT] RECORD	VO	N			N
	[INTO <i>id-1</i>] [: AT END <i>imperative-stmf</i>]	V	S			N
	<ul style="list-style-type: none"> Format 2: random retrieval READ <i>file-name-1</i> RECORD [INTO <i>id-1</i>] [: INVALID KEY <i>imperative-stmf</i>] 	V	S			N
	<ul style="list-style-type: none"> Format 3: direct I-O WITH PROTECT READ <i>file-name-1</i> [NEXT] RECORD [INTO <i>id-1</i>] 	V	S			N
	[WITH [NO] PROTECT]	VO	N			N
	[: INVALID KEY <i>imperative-stmf</i>]	V	S			N

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
RECEIVE	<ul style="list-style-type: none"> Format RECEIVE <i>cd-name</i> <MESSAGE SEGMENT> INTO <i>id-1</i> [; NO DATA <i>imperative-stmt</i>] 	N	N			N
RELEASE	<ul style="list-style-type: none"> Format RELEASE <i>record-name-1</i> [FROM <i>id-1</i>] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
RETURN	<ul style="list-style-type: none"> Format RETURN <i>file-name-1</i> RECORD [INTO <i>id-1</i>] ; AT END <i>imperative-stmt</i> 	V	S	<i>file-name-1</i>	<i>id-1</i>	N
RE-WRITE	<ul style="list-style-type: none"> Format 1: sequential files REWRITE <i>record-name</i> [FROM <i>id</i>] 	V	S	<i>record-name</i> <i>id-1</i>	<i>record-name</i>	N
	<ul style="list-style-type: none"> Format 2: indexed and relative files REWRITE <i>record-name</i> [FROM <i>id</i>] [; INVALID KEY <i>imperative-stmt</i>] 	V	S	<i>record-name</i> <i>id-1</i>	<i>record-name</i>	N
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [; AT END <i>imperative-stmt-1</i>] ; WHEN <i>condition-1</i> <<i>imperative-stmt-2</i> NEXT SENTENCE> 	V	S	<i>id-1</i> <i>condition-1</i>	<i>id-2</i>	N
	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [; AT END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> <IS EQUAL TO IS = > <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> <IS EQUAL TO IS = > <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] <<i>imperative-stmt-2</i> NEXT SENTENCE> 	V	S	<i>id-1</i> <i>data-name-1</i>		N
SEND	<ul style="list-style-type: none"> Format 1 SEND <i>cd-name</i> FROM <i>id-1</i> 	N	N			N
	<ul style="list-style-type: none"> Format 2 SEND <i>cd-name</i> [FROM <i>id-1</i>] WITH <<i>id-2</i> ESI EMI EGI> [<BEFORE AFTER> ADVANCING <<<i>id-3</i> <i>integer</i>> [LINE LINES] <i>mnemonic-name</i> PAGE>] 	N	N			N

1-114 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SET	<ul style="list-style-type: none"> Format 1: external switches SET <i>mnemonic-name-1</i> TO <ON OFF> 	V	S		<i>mnemonic-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: basic table handling SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-2</i> <i>integer-1</i>> 	V	S	<i>index-name-2</i> <i>id-2</i>	<i>index-name-1</i> <i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3: adjusting indexes SET <i>index-name-3</i> <UP BY DOWN BY> <<i>id-3</i> <i>integer-2</i>> 	V	S	<i>index-name-3</i> <i>id-3</i>	<i>index-name-3</i>	N
	<ul style="list-style-type: none"> Format 4: asynchronous processing SET RANK FOR <<i>area-name</i> <i>section-name</i>> <UP BY DOWN BY TO> <<i>id-1</i> <i>literal-1</i>> 	N	N			N
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> ON <ASCENDING DESCENDING> KEY <i>data-name-1</i> [COLLATING SEQUENCE IS <i>alphabet-name-1</i>] <USING <i>file-name-2</i> INPUT PROCEDURE IS <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>> <GIVING <i>file-name-3</i> OUTPUT PROCEDURE IS <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i>> 	V	S	<i>file-name-1</i> <i>data-name-1</i> <i>file-name-2</i>	<i>file-name-3</i>	N
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [KEY IS <EQUAL TO = GREATER THAN > NOT LESS [THAN] NOT < > <i>data-name-1</i>] [; INVALID KEY <i>imperative-stmt</i>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N
STRING	<ul style="list-style-type: none"> Format STRING <<i>id-1</i> <i>literal-1</i>> DELIMITED BY <<i>id-2</i> <i>literal-2</i> SIZE> INTO <i>id-3</i> [WITH POINTER <i>id-4</i>] [; ON OVERFLOW <i>imperative-stmt</i>] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>	<i>id-3</i> <i>id-4</i>	S

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <id-1 literal-1> FROM id-2 [ROUNDED] [: ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <id-1 literal-1> FROM <id-2 literal-2> GIVING id-3 [ROUNDED] [: ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> id-1 FROM id-2 [ROUNDED] [: ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-2	S
SUPPRESS	<ul style="list-style-type: none"> Format SUPPRESS PRINTING 	V	S			N
TERMINATE	<ul style="list-style-type: none"> Format TERMINATE report-name-1 	V	S	report-name-1		N
TRANSFORM	<ul style="list-style-type: none"> Format TRANSFORM id-3 CHARACTERS FROM <figurative-constant-1 nonnumeric-literal-1 id-1> TO <figurative-constant-2 non-numeric-literal-2 id-2> 	VO	N			N
UNLOCK	<ul style="list-style-type: none"> Format UNLOCK id-1 	N	N			N
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING id-1 [DELIMITED BY [ALL] <id-2 literal-1> [, OR [ALL] <id-3 literal-2>] INTO id-4 [, DELIMITER IN id-5] [, COUNT IN id-6] [WITH POINTER id-7] [TALLYING IN id-8] [: ON OVERFLOW imperative-stmf] 	V	S	id-1 id-2 id-3 id-4 id-5 id-6 id-7 id-8	id-4 id-5 id-6 id-7 id-8	S

1-116 Supported COBOL Statements
Supported Unisys ASCII COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
USE	<ul style="list-style-type: none"> Format 1: file I-O USE AFTER STANDARD <EXCEPTION ERROR> PROCEDURE ON <file-name INPUT OUTPUT I-O EXTEND> 	V	N			N
	<ul style="list-style-type: none"> Format 2: sequential file I-O USE <BEFORE AFTER> STANDARD [BEGINNING ENDING] [REEL FILE] LABEL PROCEDURE ON <file-name INPUT OUTPUT I-O> 	VO	N			N
	<ul style="list-style-type: none"> Format 3: report writer USE BEFORE REPORTING <i>id</i> 	VO	N			N
	<ul style="list-style-type: none"> Format 4: debug mode <i>section-name</i> SECTION [<i>segment-number</i>] USE FOR DEBUGGING ON <<i>cd-name-1</i> [ALL REFERENCES OF] <i>id-1</i> <i>file-name-1</i> <i>procedure-name-1</i> ALL PROCEDURES> 	VO	N			N
	<ul style="list-style-type: none"> Format 5: table error USE FOR TABLE ERROR REFERENCES ON <ALL <i>id-1</i>> 	N	N			N
	<ul style="list-style-type: none"> Format 6: asynchronous processing USE FOR RANDOM PROCESSING [USING <<i>file-name-1</i> <i>cd-name-1</i>>] [WITH RANK OF <i>integer-1</i>] [FOR <i>integer-2</i> CYCLES] 	VO	N			N
WRITE	<ul style="list-style-type: none"> Format 1: sequential WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [<BEFORE AFTER> ADVANCING <<<i>id-1</i> <i>integer-1</i>> [LINE LINES] <i>mnemonic-name</i> PAGE>] [; AT <END-OF-PAGE EOP> <i>imperative-stm</i>] 	V	S	<i>record-name-1</i> <i>id-1</i> <i>id-2</i>	<i>record-name-1</i>	N
		VO	N			N
		V	S			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [; INVALID KEY <i>imperative-stm</i>] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N

Supported Unisys UCS COBOL statements

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S
NULL / NULLS	V	S

Special registers

Constants	Parser	CE
DEBUG ITEM	V	N
LINAGE-COUNTER	V	N
LINE-COUNTER	V	N
PAGE-COUNTER	V	N

1-118 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Literals

Literal type	Format	Parser	CE
Nonnumeric literal	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
Numeric literals	<ul style="list-style-type: none"> Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format <i>text-name-1</i> [<IN OF> <i>library-name-1</i>] 	V	S
to Procedure Division	<ul style="list-style-type: none"> Format <i>paragraph-name-1</i> [<IN OF> <i>section-name-1</i>] 	V	S
to Data Division	<ul style="list-style-type: none"> Format 1: subscripting <i>data-name-1</i> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) (<i>leftmost-character-position: [length]</i>) 	V	S
	<ul style="list-style-type: none"> Format 2 <<i>condition-name-1</i> <i>data-name-1</i>> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 3 LINAGE-COUNTER [<IN OF> <i>file-name-2</i>] 	V	S
	<ul style="list-style-type: none"> Format 4 <PAGE-COUNTER LINE-COUNTER> <IN OF> <i>report-name</i> 	V	S

Division	Format	Parser	CE
Condition names	<ul style="list-style-type: none"> Format: Data Division <i>condition-name-1</i> [<IN OF> <i>data-name-1</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) 	V	S
Subscript	<ul style="list-style-type: none"> Format <<i>integer-1</i> ALL <i>data-name-3</i> [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>> 	V	S
Reference modification	<ul style="list-style-type: none"> Format <<i>data-name-1</i>> (<i>leftmost-character-position</i>: [<i>length</i>]) 	V	S

Control Division

Names	Format	Parser	CE
Control Division Entry	<ul style="list-style-type: none"> Format [CONTROL DIVISION]. [ALPHABET SECTION]. [SOURCE-ALPHABET CHARACTERS ARE <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]] 	N	N
Inline Section	<ul style="list-style-type: none"> Format APPLY INLINE <ADD ALL CONVERSION IF INSPECT MOVE SIGN> FOR <ALL PROCEDURES <i>procedure-name-1</i>> [<THROUGH THRU> <i>procedure-name-2</i>] 	N	N

1-120 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Identification Division

Names	Format	Parser	CE
Identification Division	<ul style="list-style-type: none"> Format IDENTIFICATION DIVISION. PROGRAM-ID <i>program-name</i>. [AUTHOR. [<i>comment-entry</i>]] [INSTALLATION. [<i>comment-entry</i>]] [DATE-WRITTEN. [<i>comment-entry</i>]] [DATE-COMPILED. [<i>comment-entry</i>]] [SECURITY. [<i>comment-entry</i>]] 	V	S
	[REMARKS. [<i>comment-entry</i>]]	VO	N

Environment Division

Input-output section

Paragraphs and clauses	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential files FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i> 	V	S
	[; RESERVE <i>integer</i> [AREA AREAS]] [; ORGANIZATION IS SEQUENTIAL] [; ACCESS MODE IS SEQUENTIAL] [FILE STATUS IS <i>data-name-1</i>].	VO	N
	<ul style="list-style-type: none"> Format 2: indexed files FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i> 	V	S
	[; RESERVE <i>integer</i> [AREA AREAS]] [; ORGANIZATION IS INDEXED] [; ACCESS MODE IS <SEQUENTIAL RANDOM DYNAMIC>] [; RECORD KEY IS <i>data-name-2</i> [; ALTERNATE RECORD KEY IS <i>data-name-3</i> [WITH DUPLICATES] [; FILE STATUS IS <i>data-name-1</i>].	VO	N

	<ul style="list-style-type: none"> • Format 3: relative files FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i> 	V	S
	[; RESERVE <i>integer</i> [AREA AREAS]] ; ORGANIZATION IS RELATIVE [; ACCESS MODE IS <SEQUENTIAL [, RELATIVE KEY IS <i>data-name-1</i>] <RANDOM DYNAMIC> RELATIVE KEY IS <i>data-name-1</i> >] [; FILE STATUS IS <i>data-name-2</i>].	VO	S
	<ul style="list-style-type: none"> • Format 4: sort-merge FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN TO <i>assignment-name-1</i>. 	V	S
	<ul style="list-style-type: none"> • Format 5: IBM tape processing FILE-CONTROL. SELECT [OPTIONAL] <i>file-name</i> ASSIGN TO <i>assignment-name</i> 	V	S
	[; RESERVE < <i>integer</i> NO> ALTERNATE [AREA AREAS]] [; ACCESS MODE IS SEQUENTIAL] [; ORGANIZATION IS SEQUENTIAL].	VO	N
I-O-CONTROL paragraph	<ul style="list-style-type: none"> • Format 1: sequential files I-O CONTROL. [; RERUN ON <<i>assignment-name-1</i> <i>file-name-1</i>> EVERY <<<i>integer-1</i> RECORDS END OF <REEL UNIT>> OF <i>file-name-1</i> <i>integer-2</i> CLOCK-UNITS <i>condition-name</i>>] [; SAME [RECORD] AREA FOR <i>file-name-3</i>] [; MULTIPLE FILE TAPE CONTAINS <i>file-name-5</i> [POSITION <i>integer-3</i>]]. 	VO	N
	<ul style="list-style-type: none"> • Format 2: indexed and relative files I-O CONTROL. [RERUN ON <<i>assignment-name-1</i> <i>file-name-1</i>> EVERY <<i>integer-1</i> RECORDS OF <i>file-name-1</i> <i>integer-2</i> CLOCK-UNITS <i>condition-name</i>>] [; SAME [RECORD] AREA FOR <i>file-name-3</i>]. 	VO	N
	<ul style="list-style-type: none"> • Format 3: sort-merge I-O CONTROL. [; SAME <RECORD SORT SORT-MERGE> AREA FOR <i>file-name-1</i>]. 	VO	N
	<ul style="list-style-type: none"> • Format 4: IBM tape processing I-O CONTROL. [; RERUN ON <i>assignment-name-1</i> EVERY <<i>integer-1</i> RECORDS [END OF] REEL> OF <i>file-name-1</i>] [; SAME [RECORD] AREA FOR <i>file-name-2</i>] 	VO	N

1-122 Supported COBOL Statements
Supported Unisys UCS COBOL statements

	<ul style="list-style-type: none"> • Format 5: direct I-O I-O CONTROL. [; APPLY INTERLOCK ON <i>file-name-1</i>] [; APPLY PROTECT ON <i>file-name-2</i> [FOR <i>integer-3</i> RECORDS]] 	VO	N
	<ul style="list-style-type: none"> • Format 6: indexed sequential I-O I-O CONTROL. [; APPLY INTERLOCK ON <i>file-name-1</i>] [; APPLY PROTECT ON <i>file-name-2</i> [FOR <i>integer-3</i> RECORDS]] [; APPLY CORE-INDEX ON <i>file-name-3</i>] 	VO	N

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COM-PUTER paragraph	<ul style="list-style-type: none"> • Format OBJECT-COMPUTER. computer-name [, MEMORY SIZE <i>integer</i> <WORDS CHARACTERS MODULES>] [, PROGRAM COLLATING SEQUENCE IS <i>alphabet-name</i>]. 	VO	N
SOURCE-COM-PUTER paragraph	<ul style="list-style-type: none"> • Format SOURCE-COMPUTER. computer-name. 	VO	N
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> • Format SPECIAL-NAMES. [, <i>implementor-name</i> <IS <i>mnemonic-name</i> [, ON STATUS IS <i>condition-name-1</i>, [OFF STATUS IS <i>condition-name-2</i>]] IS <i>mnemonic-name</i> [, OFF STATUS IS <i>condition-name-2</i>, [ON STATUS IS <i>condition-name-1</i>]] ON STATUS IS <i>condition-name-1</i>, [OFF STATUS IS <i>condition-name-2</i>] OFF STATUS IS <i>condition-name-2</i>, [ON STATUS IS <i>condition-name-1</i>]->] 	V	PS
	[, <i>alphabet-name</i> IS <STANDARD-1 NATIVE <i>implementor-name</i> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i>]->]	VO	N
	[, CURRENCY SIGN IS <i>literal-4</i>]	VO	N
	[, DECIMAL POINT IS COMMA]	V	S
	[, CLASS NAME. IS [, <i>mnemonic-name</i>] [VALUE IS <i>literal-5</i> [<THROUGH THRU> <i>literal-6</i>]]]	VO	N
ACTUAL KEY clause	<ul style="list-style-type: none"> • Format ACTUAL KEY IS <i>data-name</i> 	VO	N

APPLY clause	<ul style="list-style-type: none"> Format APPLY <EXREF EXDEF> ON <i>id-1</i> 	VO	N
APPLY WRITE-ONLY clause	<ul style="list-style-type: none"> Format APPLY WRITE-ONLY ON <i>file-name-1</i> 	VO	N
ASSIGN clause	<ul style="list-style-type: none"> Format ASSIGN TO <i>assignment-name</i> 	V	S
FILE-LIMITS clause	<ul style="list-style-type: none"> Format [: <FILE-LIMIT IS FILE-LIMITS ARE> <i><data-name-1 literal-1></i> <THROUGH THRU> <i><data-name-2 literal-2></i>] [: <i><data-name-3 literal-3></i> <THROUGH THRU> <i><data-name-4 literal-4></i>] 	VO	N
PROCESSING MODE clause	<ul style="list-style-type: none"> Format PROCESSING MODE IS SEQUENTIAL 	VO	N
RERUN clause	<ul style="list-style-type: none"> Format RERUN [ON <i><file-name-1 UNISERVO UNISERVOS MASS-STORAGE></i>] EVERY <<<i>integer-1</i> RECORDS END OF <REEL UNIT>> OF <i>file-name-1</i> 	VO	N
RESERVE clause	<ul style="list-style-type: none"> Format RESERVE <<i>integer-1</i> NO> ALTERNATE [AREA AREAS] 	VO	N
SEGMENT-LIMIT clause	<ul style="list-style-type: none"> Format SEGMENT LIMIT IS <i>segment-number</i> 	VO	N
WITH DEBUGGING MODE clause	<ul style="list-style-type: none"> Format WITH DEBUGGING MODE 	VO	N
WITH RANK clause	<ul style="list-style-type: none"> Format WITH RANK OF <i>integer</i> 	VO	N

1-124 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Data Division

Entries and Clauses	Format	Parser	CE
Communication Description Entry	<ul style="list-style-type: none"> Format 1 CD <i>cd-name</i>; FOR [INITIAL] INPUT [; SYMBOLIC QUEUE IS <i>data-name-1</i>] [; SYMBOLIC SUB-QUEUE-1 IS <i>data-name-2</i>] [; SYMBOLIC SUB-QUEUE-2 IS <i>data-name-3</i>] [; SYMBOLIC SUB-QUEUE-3 IS <i>data-name-4</i>] [; MESSAGE DATE IS <i>data-name-5</i>] [; MESSAGE TIME IS <i>data-name-6</i>] [; SYMBOLIC SOURCE IS <i>data-name-7</i>] [; TEXT LENGTH IS <i>data-name-8</i>] [; END KEY IS <i>data-name-9</i>] [; STATUS KEY IS <i>data-name-10</i>] [; MESSAGE COUNT IS <i>data-name-11</i>] 	N	N
	<ul style="list-style-type: none"> Format 2 CD <i>cd-name</i>; FOR OUTPUT [; DESTINATION COUNT IS <i>data-name-1</i>] [; TEXT LENGTH IS <i>data-name-2</i>] [; STATUS KEY IS <i>data-name-3</i>] [; DESTINATION TABLE OCCURS <i>integer-2</i> TIMES [; INDEXED BY <i>index-name-1</i>]] [; ERROR KEY IS <i>data-name-4</i>] [; SYMBOLIC DESTINATION IS <i>data-name-5</i>] 	N	N
Data Description Entry	<ul style="list-style-type: none"> Format 1 <i>level-number</i> <<i>data-name-1</i> FILLER> [; REDEFINES <i>data-name-2</i>] [; <PICTURE PIC> IS <i>character-string</i>] [[USAGE IS] <COMPUTATIONAL COMP DISPLAY COMPUTATIONAL-1 COMP-1 COMPUTATIONAL-2 COMP-2 DISP>] [; [SIGN IS] <LEADING TRAILING> [SEPARATE CHARACTER]] [; <SYNCHRONIZED SYNC> [LEFT RIGHT]] [; <JUSTIFIED JUST> RIGHT] [; BLANK WHEN ZERO] [; VALUE IS <i>literal</i>]. 	V	S
	<ul style="list-style-type: none"> Format 2 66 <i>data-name-1</i>; RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>]. 	V	S
	<ul style="list-style-type: none"> Format 3 88 <i>condition-name</i>; <VALUE IS VALUES ARE> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>] [, <i>literal-3</i> [<THROUGH THRU> <i>literal-4</i>]]. 	V	S

Entries and Clauses	Format	Parser	CE
File Description (FD) Entry	<ul style="list-style-type: none"> Format 1: sequential files <p><i>FD file-name</i></p>	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; DATA <RECORD IS RECORDS ARE> <i>data-name-2</i>] [; LINAGE IS <<i>data-name-3</i> <i>integer-5</i>> LINES [, WITH FOOTING AT <<i>data-name-4</i> <i>integer-6</i>>] [, LINES AT TOP <<i>data-name-6</i> <i>integer-7</i>>] [, LINES AT BOTTOM <<i>data-name-7</i> <i>integer-8</i>>]] [; CODE-SET IS <i>alphabet-name</i>] DENSITY IS <6 8 12> LINES.</pre>	VO	N
	<ul style="list-style-type: none"> Format 2: indexed and relative files <p><i>FD file-name</i></p>	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-2</i>].</pre>	VO	N
	<ul style="list-style-type: none"> Format 3: IBM tape processing <p><i>FD file-name</i></p>	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; DATA <RECORD [IS] RECSORDS [ARE]> <i>data-name-2</i>]. ; RECORDING MODE IS <E U V> [AN]</pre>	VO	N
	<ul style="list-style-type: none"> Format 4: sort-merge <p><i>SD file-name</i></p>	V	S
	<pre>[; RECORD CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> CHARACTERS] [; DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-1</i>].</pre>	VO	N

1-126 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Entries and Clauses	Format	Parser	CE
	<ul style="list-style-type: none"> Format 5: report writer FD <i>file-name</i> 	V	S
	<pre>[; BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [; RECORD CONTAINS [<i>integer-3</i> TO] <i>integer-4</i> CHARACTERS] ; LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> [; VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>>] [; CODE-SET IS <i>alphabet-name</i>] ; <REPORT IS REPORTS ARE> <i>report-name</i> DENSITY IS <6 8 12> LINES.</pre>	VO	N
Report Description (RD) Entry	<ul style="list-style-type: none"> Format RD <i>report-name</i> [; CODE <i>literal-1</i>] [; <CONTROL IS CONTROLS ARE> <<i>data-name-1</i> FINAL [, <i>data-name-1</i>]>] [; PAGE [LIMIT IS LIMITS ARE] <i>integer-1</i> [LINE LINES] [, HEADING <i>integer-2</i>] [, FIRST DETAIL <i>integer-3</i>] [, LAST DETAIL <i>integer-4</i>] [, FOOTING <i>integer-5</i>]. 	V	S
Report Group Description	<ul style="list-style-type: none"> Format 1 01 [<i>data-name-1</i>] [; LINE NUMBER IS <<i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i>>] [; NEXT GROUP IS <<i>integer-3</i> PLUS <i>integer-4</i> NEXT PAGE>] [; TYPE IS <REPORT HEADING RH PAGE HEADING PH <CONTROL HEADING CH> <<i>data-name-2</i> FINAL> DETAIL DE <CONTROL FOOTING CF> <<i>data-name-3</i> FINAL> PAGE FOOTING PF REPORT FOOTING RF> [; [USAGE IS] <DISPLAY DISP>] Format 2 <i>level-number</i> [<i>data-name-1</i>] [; LINE NUMBER IS <<i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i>>] [; [USAGE IS] <DISPLAY DISP>] Format 3 <i>level-number</i> [<i>data-name-1</i>] [; BLANK WHEN ZERO] [; GROUP INDICATE] [; <JUSTIFIED JUST> RIGHT] [; LINE NUMBER IS <<i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i>>] [; COLUMN NUMBER IS <i>integer-3</i>] [; <PICTURE PIC> IS <i>character-string</i> ; SOURCE IS <i>id-1</i> < ; VALUE IS <i>literal</i> ; SUM <i>id-2</i> [UPON <i>data-name-2</i>]> [; RESET ON <<i>data-name-3</i> FINAL>] [; [USAGE IS] <DISPLAY DISP>] 	V	S
Saved-Area Description	<ul style="list-style-type: none"> Format SA <i>area-name</i> ; AREA CONTAINS <i>integer-1</i> RECORDS [; WITH RANK OF <i>integer-2</i>] 	N	N

Entries and Clauses	Format	Parser	CE
AREA CONTAINS clause	<ul style="list-style-type: none"> Format AREA CONTAINS <i>integer-1</i> RECORDS 	N	N
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> Format BLANK WHEN ZERO 	V	S
BLOCK CONTAINS clause	<ul style="list-style-type: none"> Format BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS> 	N	N
CODE clause	<ul style="list-style-type: none"> Format CODE <i>literal-1</i> 	V	N
CODE-SET clause	<ul style="list-style-type: none"> Format CODE-SET IS <i>alphabet-name</i> 	V	N
COLUMN NUMBER clause	<ul style="list-style-type: none"> Format COLUMN NUMBER IS <i>integer-1</i> 	V	N
CONTROL clause	<ul style="list-style-type: none"> Format <CONTROL IS CONTROLS ARE> <i><data-name-1 FINAL></i> 	V	N
Data-name or FILLER clause	<ul style="list-style-type: none"> Format <<i>data-name</i> FILLER> 	V	S
DATA RECORDS clause	<ul style="list-style-type: none"> Format DATA <RECORD IS RECORDS ARE> <i>data-name</i> 	V	S
DENSITY clause	<ul style="list-style-type: none"> Format DENSITY IS <6 8 12> LINES 	VO	N
JUSTIFIED clause	<ul style="list-style-type: none"> Format <JUSTIFIED JUST> RIGHT 	V	S
LABEL RECORDS clause	<ul style="list-style-type: none"> Format LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED> 	V	S

1-128 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Entries and Clauses	Format	Parser	CE
LINAGE clause	<ul style="list-style-type: none"> Format LINAGE IS <data-name-1 integer-1> LINES [, WITH FOOTING AT <data-name-2 integer-2>] [, LINES AT TOP <data-name-3 integer-3>] [, LINES AT BOTTOM <data-name-4 integer-4>] 	N	N
LINE NUMBER clause	<ul style="list-style-type: none"> Format LINE NUMBER IS <integer-1 [ON NEXT PAGE] PLUS integer-2> 	V	N
NEXT GROUP clause	<ul style="list-style-type: none"> Format NEXT GROUP IS <integer-1 PLUS integer-2 NEXT PAGE> 	V	N
OCCURS clause	<ul style="list-style-type: none"> Format 1: fixed-length tables OCCURS integer-2 TIMES [<ASCENDING DESCENDING> KEY IS data-name-2] [[INDEXED BY index-name-1] 	V	S
	<ul style="list-style-type: none"> Format 2: variable-length tables OCCURS integer-1 TO integer-2 TIMES DEPENDING ON data-name-1 [<ASCENDING DESCENDING> KEY IS data-name-2] [[INDEXED BY index-name-1] 	V	S
PAGE clause	<ul style="list-style-type: none"> Format PAGE [LIMIT IS LIMITS ARE] integer-1 [LINE LINES] [, HEADING integer-2] [, FIRST DETAIL integer-3] [, LAST DETAIL integer-4] [, FOOTING integer-5] 	V	N
PICTURE clause	<ul style="list-style-type: none"> Format <PICTURE PIC> IS character-string 	V	S
POINT LOCATION clause	<ul style="list-style-type: none"> Format POINT LOCATION IS <LEFT RIGHT> integer-1 <PLACE PLACES> 	N	N
RECORD CONTAINS clause	<ul style="list-style-type: none"> Format RECORD CONTAINS integer-1 TO integer-2 CHARACTERS 	VO	N

Entries and Clauses	Format	Parser	CE
RECORDING MODE clause	<ul style="list-style-type: none"> Format 1: IBM tape processing RECORDING MODE IS <E U V> [AN] 	V	N
	<ul style="list-style-type: none"> Format 2: sequential I-O RECORDING MODE IS <<LION CFH> [AN] <FORMO1 FORMO2 FORMO3> [U] BLANK [SIGN] COMPACT INTERNAL SIGN SDF> 	V	N
	<ul style="list-style-type: none"> Format 3: direct I-O RECORDING MODE IS <BLANK [SIGN] COMPACT INTERNAL SIGN SDF> 	V	N
	<ul style="list-style-type: none"> Format 4: Information Interchange tape processing RECORDING MODE IS <D F S U> 	V	N
REDEFINES clause	<ul style="list-style-type: none"> Format <i>level-number data-name-1; REDEFINES data-name-2</i> 	V	S
RENAMES clause	<ul style="list-style-type: none"> Format <i>66 data-name-1; RENAMES data-name-2 [<THROUGH THRU> data-name-3].</i> 	V	S
REPORT clause	<ul style="list-style-type: none"> Format <REPORT IS REPORTS ARE> <i>report-name</i> 	V	O
SIGN clause	<ul style="list-style-type: none"> Format [SIGN IS] <LEADING TRAILING> [SEPARATE CHARACTER] 	V	S
SOURCE clause	<ul style="list-style-type: none"> Format SOURCE IS <i>id-1</i> 	V	N
SUM clause	<ul style="list-style-type: none"> Format SUM <i>id-1</i> [UPON <i>data-name-2</i>] [RESET ON <<i>data-name-3</i> FINAL>] 	V	N
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZE SYNC> [LEFT RIGHT] 	V	N
TYPE clause	<ul style="list-style-type: none"> Format TYPE IS <REPORT HEADING RH PAGE HEADING PH <CONTROL HEADING CH> <<i>data-name-2</i> FINAL> DETAIL DE <CONTROL FOOTING CF> <<i>data-name-3</i> FINAL> PAGE FOOTING PF REPORT FOOTING RF> 	V	N

1-130 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Entries and Clauses	Format	Parser	CE
USAGE clause	<ul style="list-style-type: none"> • Format 1: general [USAGE IS] <COMPUTATIONAL COMP DISPLAY COMPUTATIONAL-1 COMP-1 COMPUTATIONAL-2 COMP-2 DISP> 	V	S
	<ul style="list-style-type: none"> • Format 2: table handling [USAGE IS] INDEX 	V	S
	<ul style="list-style-type: none"> • Format 3: asynchronous processing [USAGE IS] LOCK 	N	N
VALUE clause	<ul style="list-style-type: none"> • Format 1 VALUE IS <i>literal</i> 	V	S
	<ul style="list-style-type: none"> • Format 2 <VALUE IS VALUES ARE> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>] [, <i>literal-3</i> [<THROUGH THRU> <i>literal-4</i>]] 	V	S
VALUE OF clause	<ul style="list-style-type: none"> • Format 1 VALUE OF <i>system-name-1</i> IS <<i>data-name-1</i> <i>literal-1</i>> 	VO	N
	<ul style="list-style-type: none"> • Format 2: sequential and direct I-O VALUE OF <<i>system-name-1</i> FILE-ID SET-ID CREATION-DATE PURGE-DATE FILE-ACCESS FILE-QUALIFIER> IS <<i>data-name-1</i> <i>literal-1</i>> 	VO	N

Procedure Division

Arithmetic operators

Binary operation	Meaning	Format	Parser	CE			
				Gen.	Read variables	Written variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

1-132 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Conditional Expressions

Conditions	Format	Parser	CE			
			Gen.	Read variables	Written variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id</i> IS [NOT] <NUMERIC ALPHABETIC CLASS-NAME <i>mnemonic-name</i>> 	V	S	<i>id-1</i>		S
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format <i>oper-1</i> IS [NOT] <GREATER THAN LESS THAN EQUAL TO> <i>oper-2</i> <i>oper-1</i>: <<i>id-1</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>oper-2</i>: <i>id-2</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> IS [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	<i>operand-1</i>		S
Switch-status condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT <i>simple-condition</i> 	V	S	<i>condition-1</i>		S
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1</i> <AND OR> <i>condition-2</i> 	V	S	<i>condition-1</i> <i>condition-2</i>		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format <i>relation-condition</i> <AND OR> [NOT] [<i>relational-operator</i>] <i>object</i> 	V	S	<i>relation-condition</i> <i>object</i>		S

Statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format PROCEDURE DIVISION [USING <i>data-name-1</i>] [WITH ENTRY POINTS <i>procedure-name-1</i>] 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N
DECLARATIVES	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. <i>sentence</i>] END DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FROM <i>mnemonic-name</i>] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE [YYYYMMDD] DAY [YYYYDDD] TIME DATE-TIME [MMDYYYY]> 	VO	N			N
	<ul style="list-style-type: none"> Format 3: ACCEPT <i>id</i> [FROM <CARD-READER CONSOLE DATE-TIME>] 	VO	N			N
ACCEPT MESSAGE COUNT	<ul style="list-style-type: none"> Format ACCEPT <i>cd-name</i> MESSAGE COUNT 	N	N			N
ADD	<ul style="list-style-type: none"> Format 1 ADD <<i>id-1</i> <i>literal-1</i>> TO <i>id-2</i> [ROUNDED] [: ON SIZE ERROR <i>imperative-stmf</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <<i>id-1</i> <i>literal-1</i>>, <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [:ON SIZE ERROR <i>imperative-stmf</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> [ROUNDED] [:ON SIZE ERROR <i>imperative-stmf</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N

1-134 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
ALL	<ul style="list-style-type: none"> Format ALL <INACTIVE ACTIVE> FOR <area-name-1 section-name-1> 	N	N			N
ALTER	<ul style="list-style-type: none"> Format ALTER <i>procedure-name-1</i> TO [PROCEED TO] <i>procedure-name-2</i> 	VO	N			N
ANY	<ul style="list-style-type: none"> Format ANY <INACTIVE ACTIVE> FOR <area-name-1 section-name-1> 	N	N			N
CALL	<ul style="list-style-type: none"> Format 1: general CALL <<i>id-1</i> <i>literal-1</i>> [USING <<i>data-name-1</i> <i>literal-2</i>>] [; ON OVERFLOW <i>imperative-stmt</i>] 	V	S	<i>id-1 data-name-1</i>		S
	<ul style="list-style-type: none"> Format 2: CALL debug CALL 'PADSS' 	V	S			N
CANCEL	<ul style="list-style-type: none"> Format CANCEL <<i>id-1</i> <i>literal-1</i>> 	V	S			N
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> [<REEL UNIT> [FOR REMOVAL WITH NO REWIND] WITH <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files CLOSE <i>file-name-1</i> [WITH LOCK] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 3: IBM tape processing CLOSE <i>file-name-1</i> [<REEL UNIT> [WITH NO REWIND] WITH <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	VO	N			N	

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <i>id-1</i> [ROUNDED] = <i>arithmetic-expr</i> [; ON SIZE ERROR <i>imperative-stmt</i>] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S
COPY	<ul style="list-style-type: none"> Format COPY <i>text-name</i> [<OF IN> <i>library-name</i>] [REPLACING, <==<i>pseudo-text-1</i>== <i>id-1</i> <i>literal-1</i> <i>word-1</i>> BY <==<i>pseudo-text-2</i>== <i>id-2</i> <i>literal-2</i> <i>word-2</i>>] 	V	S			N
DELETE	<ul style="list-style-type: none"> Format: indexed and relative files DELETE <i>file-name</i> RECORD [; INVALID KEY <i>imperative-stmt</i>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISABLE	<ul style="list-style-type: none"> Format DISABLE <INPUT [TERMINAL] OUTPUT> <i>cd-name</i> WITH KEY <<i>id-1</i> <i>literal-1</i>> 	N	N			N
DISPLAY	<ul style="list-style-type: none"> Format 1 DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <CARD-PUNCH CONSOLE PRINTER>] 	V	S	<i>id-1</i>		S
		VO	N			N
	<ul style="list-style-type: none"> Format 2 DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <i>mnemonic-name</i>] 	V	S	<i>id-1</i>		S

1-136 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <id-1 literal-1> INTO id-2 [ROUNDED] [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3 id-4	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [; ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3 id-4	S
ENABLE	<ul style="list-style-type: none"> Format ENABLE <INPUT [TERMINAL] OUTPUT> cd-name WITH KEY <id-1 literal-1> 	V	S	cd-name id-1		N
ENTER	<ul style="list-style-type: none"> Format 1 ENTER <MASM ASM PL1 FTN> literal [USING argument] 	V	S	argument	argument	N
	<ul style="list-style-type: none"> Format 2 ENTER MASM "C\$INFO" USING file-name, data-name 	V	S	file-name data-name	file-name data-name	N
	<ul style="list-style-type: none"> Format 3 ENTER <FORTRAN FD [ASM]> literal [USING argument] 	V	S	argument	argument	N

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
EXAMINE	<ul style="list-style-type: none"> Format EXAMINE <i>id-1</i> <TALLYING <UNTIL FIRST ALL LEADING TRAILING> <<i>id-2</i> <i>literal-1</i>> [REPLACING BY <<i>id-3</i> <i>literal-2</i>>] REPLACING <[UNTIL] FIRST ALL LEADING TRAILING> <<i>id-2</i> <i>literal-1</i>> BY <<i>id-3</i> <i>literal-2</i>>> 	V	S	<i>id-1</i>		N
EXHIBIT	<ul style="list-style-type: none"> Format EXHIBIT <NAMED CHANGED CHANGED NAMED> <<i>data-name-1</i> <i>literal-1</i>> 	V	S	<i>data-name-1</i>		N
EXIT	<ul style="list-style-type: none"> Format <i>paragraph-name</i>. EXIT. 	V	S			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
FREE	<ul style="list-style-type: none"> Format FREE <i>file-name</i> RECORD 	V	S			N
GENERATE	<ul style="list-style-type: none"> Format GENERATE <<i>data-name</i> <i>report-name</i>> 	V	S	<i>data-name</i>		N
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO TO <i>procedure-name-1</i> 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO TO <i>procedure-name-1</i> DEPENDING ON <i>id-1</i> 	V	S	<i>id-1</i>		N
HOLD	<ul style="list-style-type: none"> Format HOLD <ALL <i>are-name-1</i> <i>section-name-1</i>> 	N	N			N
IF	<ul style="list-style-type: none"> Format IF <i>condition</i>; <<i>stmt-1</i> NEXT SENTENCE>; ELSE <<i>stmt-2</i> NEXT SENTENCE> 	V	S	<i>condition-1</i>		N
INITIATE	<ul style="list-style-type: none"> Format INITIATE <i>report-name-1</i> 	V	S		<i>report-name-1</i>	N

1-138 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING , <i>id-2</i> FOR, <CHARACTERS <TRAILING ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] , <TRAILING ALL LEADING FIRST> , <<i>id-5</i> <i>literal-3</i>> BY <<i>id-7</i> <i>literal-5</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-5</i> <i>id-6</i> <i>id-7</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING , <i>id-2</i> FOR, <CHARACTERS <TRAILING ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] , <TRAILING ALL LEADING FIRST> , <<i>id-5</i> <i>literal-3</i>> BY <<i>id-7</i> <i>literal-5</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i> <i>id-7</i>	<i>id-1</i>	N
LOCK	<ul style="list-style-type: none"> Format LOCK <i>id-1</i> 	N	N			N
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> ON <ASCENDING DESCENDING> KEY <i>data-name-1</i> [COLLATING SEQUENCE IS <i>alphabet-name-1</i>] 	V	S	<i>file-name-1</i> <i>data-name-1</i>	<i>file-name-4</i>	N
	<ul style="list-style-type: none"> USING <i>file-name-2</i> <i>file-name-3</i> <OUTPUT PROCEDURE IS <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i>> 	VO	N	<i>file-name-2</i> <i>file-name-3</i>		N
	<ul style="list-style-type: none"> Format MONITOR <<i>procedure-name-1</i> <i>id-1</i> ALL> [UNTIL <i>procedure-name</i>] 	V	S			N
MONITOR	<ul style="list-style-type: none"> Format MONITOR <<i>procedure-name-1</i> <i>id-1</i> ALL> [UNTIL <i>procedure-name</i>] 	VO	N			N

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <id-1 literal-1> TO id-2 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> id-1 TO id-2 	V	S	id-1	id-2	N
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <id-1 literal-1> BY id-2 [ROUNDED] [: ON SIZE ERROR imperative-stmt] 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [: ON SIZE ERROR imperative-stmt] 	V	S	id-1 id-2	id-3	S
NOTE	<ul style="list-style-type: none"> Format NOTE character-string. 	VO	N			N
ON	<ul style="list-style-type: none"> Format ON integer-1 [AND EVERY integer-2] [UNTIL integer-3] ; <imperative-stmt-1 NEXT SENTENCE> [: <OTHERWISE ELSE <imperative-stmt-2 NEXT SENTENCE>] 	V	S			N
OPEN	<ul style="list-style-type: none"> Format 1: sequential files OPEN <INPUT file-name-1 [REVERSED WITH NO REWIND] OUTPUT file-name-2 [WITH NO REWIND] I-O file-name-3 EXTEND file-name-4> 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
		VO	N			N
		V	S			N
		VO	N			N
		V	S			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files OPEN <INPUT file-name-1 OUTPUT file-name-2 I-O file-name-3> 	V	S			file-name-1 file-name-2 file-name-3

1-140 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <i><id-1 integer-1></i> TIMES 	V	S	<i>id-1</i>		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] UNTIL <i>condition-1</i> 	V	S	<i>condition-1</i>		N
	<ul style="list-style-type: none"> Format 4: with VARYING phrase PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] VARYING <i><id-2 index-name-1></i> FROM <i><id-3 index-name-2 literal-1></i> BY <i><id-4 literal-2></i> UNTIL <i>condition-1</i> [AFTER <i><id-5 index-name-3></i> FROM <i><id-6 index-name-4 literal-3></i> BY <i><id-7 literal-4></i> UNTIL <i>condition-2</i>] 	V	S	<i>id-3</i> <i>id-4</i> <i>condition-1</i> <i>id-6</i> <i>id-7</i> <i>condition-2</i>	<i>id-2</i> <i>id-5</i>	N
PROCESS	<ul style="list-style-type: none"> Format PROCESS <i>section-name</i> [FROM <i>id</i> [USING <i>area-name</i>]] 	N	N			N
READ	<ul style="list-style-type: none"> Format 1: sequential retrieval READ <i>file-name-1</i> 	V	S	<i>file-rec-1</i> (corresponding to <i>file-name-1</i>)	<i>file-rec-1</i> (corresponding to <i>file-name-1</i>) <i>id-1</i>	N
	[NEXT] RECORD	VO	N			N
	[INTO <i>id-1</i>] [: AT END <i>imperative-stmf</i>]	V	S			N
	<ul style="list-style-type: none"> Format 2: random retrieval READ <i>file-name-1</i> RECORD [INTO <i>id-1</i>] [: INVALID KEY <i>imperative-stmf</i>] 	V	S			N
	<ul style="list-style-type: none"> Format 3: direct I-O WITH PROTECT READ <i>file-name-1</i> [NEXT] RECORD [INTO <i>id-1</i>] 	V	S			N
	[WITH [NO] PROTECT]	VO	N			N
	[: INVALID KEY <i>imperative-stmf</i>]	V	S			N

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
RECEIVE	<ul style="list-style-type: none"> Format RECEIVE <i>cd-name</i> <MESSAGE SEGMENT> INTO <i>id-1</i> [; NO DATA <i>imperative-stmt</i>] 	N	N			N
RELEASE	<ul style="list-style-type: none"> Format RELEASE <i>record-name-1</i> [FROM <i>id-1</i>] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
RETURN	<ul style="list-style-type: none"> Format RETURN <i>file-name-1</i> RECORD [INTO <i>id-1</i>] ; AT END <i>imperative-stmt</i> 	V	S	<i>file-name-1</i>	<i>id-1</i>	N
REWRITE	<ul style="list-style-type: none"> Format 1: sequential files REWRITE <i>record-name</i> [FROM <i>id</i>] 	V	S	<i>record-name</i> <i>id-1</i>	<i>record-name</i>	N
	<ul style="list-style-type: none"> Format 2: indexed and relative files REWRITE <i>record-name</i> [FROM <i>id</i>] [; INVALID KEY <i>imperative-stmt</i>] 	V	S	<i>record-name</i> <i>id-1</i>	<i>record-name</i>	N
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [; AT END <i>imperative-stmt-1</i>] ; WHEN <i>condition-1</i> <<i>imperative-stmt-2</i> NEXT SENTENCE> 	V	S	<i>id-1</i> <i>condition-1</i>	<i>id-2</i>	N
	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [; AT END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> <IS EQUAL TO IS = > <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> <IS EQUAL TO IS = > <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] <<i>imperative-stmt-2</i> NEXT SENTENCE> 	V	S	<i>id-1</i> <i>data-name-1</i>		N
SEND	<ul style="list-style-type: none"> Format 1 SEND <i>cd-name</i> FROM <i>id-1</i> 	N	N			N
	<ul style="list-style-type: none"> Format 2 SEND <i>cd-name</i> [FROM <i>id-1</i>] WITH <<i>id-2</i> ESI EMI EGI> [<BEFORE AFTER> ADVANCING <<<i>id-3</i> <i>integer</i>> [LINE LINES] <i>mnemonic-name</i> PAGE>] 	N	N			N

1-142 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SET	<ul style="list-style-type: none"> Format 1: external switches SET <i>mnemonic-name-1</i> TO <ON OFF> 	V	S		<i>mnemonic-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: basic table handling SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-2</i> <i>integer-1</i>> 	V	S	<i>index-name-2</i> <i>id-2</i>	<i>index-name-1</i> <i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3: adjusting indexes SET <i>index-name-3</i> <UP BY DOWN BY> <<i>id-3</i> <i>integer-2</i>> 	V	S	<i>index-name-3</i> <i>id-3</i>	<i>index-name-3</i>	N
	<ul style="list-style-type: none"> Format 4: asynchronous processing SET RANK FOR <<i>area-name</i> <i>section-name</i>> <UP BY DOWN BY TO> <<i>id-1</i> <i>literal-1</i>> 	N	N			N
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> ON <ASCENDING DESCENDING> KEY <i>data-name-1</i> [COLLATING SEQUENCE IS <i>alphabet-name-1</i>] <USING <i>file-name-2</i> INPUT PROCEDURE IS <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <GIVING <i>file-name-3</i> OUTPUT PROCEDURE IS <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i>]> 	V	S	<i>file-name-1</i> <i>data-name-1</i> <i>file-name-2</i>	<i>file-name-3</i>	N
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [KEY IS <EQUAL TO = GREATER THAN > NOT LESS [THAN] NOT < > <i>data-name-1</i>] [; INVALID KEY <i>imperative-stmt</i>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N
STRING	<ul style="list-style-type: none"> Format STRING <<i>id-1</i> <i>literal-1</i>> DELIMITED BY <<i>id-2</i> <i>literal-2</i> SIZE> INTO <i>id-3</i> [WITH POINTER <i>id-4</i>] [; ON OVERFLOW <i>imperative-stmt</i>] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>	<i>id-3</i> <i>id-4</i>	S

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <id-1 literal-1> FROM id-2 [ROUNDED] [: ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <id-1 literal-1> FROM <id-2 literal-2> GIVING id-3 [ROUNDED] [: ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> id-1 FROM id-2 [ROUNDED] [: ON SIZE ERROR imperative-stmf] 	V	S	id-1 id-2	id-2	S
SUPPRESS	<ul style="list-style-type: none"> Format SUPPRESS PRINTING 	V	S			N
TERMINATE	<ul style="list-style-type: none"> Format TERMINATE report-name-1 	V	S	report-name-1		N
TRANSFORM	<ul style="list-style-type: none"> Format TRANSFORM id-3 CHARACTERS FROM <figurative-constant-1 nonnumeric-literal-1 id-1> TO <figurative-constant-2 non-numeric-literal-2 id-2> 	VO	N			N
UNLOCK	<ul style="list-style-type: none"> Format UNLOCK id-1 	N	N			N
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING id-1 [DELIMITED BY [ALL] <id-2 literal-1> [, OR [ALL] <id-3 literal-2>] INTO id-4 [, DELIMITER IN id-5] [, COUNT IN id-6] [WITH POINTER id-7] [TALLYING IN id-8] [: ON OVERFLOW imperative-stmf] 	V	S	id-1 id-2 id-3 id-4 id-5 id-6 id-7 id-8	id-4 id-5 id-6 id-7 id-8	S

1-144 Supported COBOL Statements
Supported Unisys UCS COBOL statements

Conditions and statements	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
USE	<ul style="list-style-type: none"> Format 1: file I-O USE AFTER STANDARD <EXCEPTION ERROR> PROCEDURE ON <file-name INPUT OUTPUT I-O EXTEND> 	V	N			N
	<ul style="list-style-type: none"> Format 2: sequential file I-O USE <BEFORE AFTER> STANDARD [BEGINNING ENDING] [REEL FILE] LABEL PROCEDURE ON <file-name INPUT OUTPUT I-O> 	VO	N			N
	<ul style="list-style-type: none"> Format 3: report writer USE BEFORE REPORTING <i>id</i> 	VO	N			N
	<ul style="list-style-type: none"> Format 4: debug mode <i>section-name</i> SECTION [<i>segment-number</i>] USE FOR DEBUGGING ON <<i>cd-name-1</i> [ALL REFERENCES OF] <i>id-1</i> <i>file-name-1</i> <i>procedure-name-1</i> ALL PROCEDURES> 	VO	N			N
	<ul style="list-style-type: none"> Format 5: table error USE FOR TABLE ERROR REFERENCES ON <ALL <i>id-1</i>> 	N	N			N
	<ul style="list-style-type: none"> Format 6: asynchronous processing USE FOR RANDOM PROCESSING [USING <<i>file-name-1</i> <i>cd-name-1</i>>] [WITH RANK OF <i>integer-1</i>] [FOR <i>integer-2</i> CYCLES] 	VO	N			N
WRITE	<ul style="list-style-type: none"> Format 1: sequential WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [<BEFORE AFTER> ADVANCING <<<i>id-1</i> <i>integer-1</i>> [LINE LINES] <i>mnemonic-name</i> PAGE>] [; AT <END-OF-PAGE EOP> <i>imperative-stmt</i>] 	V	S	<i>record-name-1</i> <i>id-1</i> <i>id-2</i>	<i>record-name-1</i>	N
		VO	N			N
		V	S			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [; INVALID KEY <i>imperative-stmt</i>] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N

Supported Unisys CDML COBOL statements

Data Division

Clauses	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DMCA	<ul style="list-style-type: none"> Format: [DMCA [AND RUN-UNIT-STATISTICS] <IS ARE> < COMMON LINKAGE WORKING > [NUMBER OF ERROR-ITEMS IS <i>int-4</i>]] 	V	N			N
ERROR	<ul style="list-style-type: none"> Format: [ERROR RECOVERY IS <i>general-error-para</i>] 	V	N			N
INVOKE	<ul style="list-style-type: none"> Format: INVOKE SUBSCHEMA <i>subschema-name</i> IN FILE <i>file-name</i> OF SCHEMA <i>schema-name</i> [KEY FOR INVOKE IS <i>lit-1</i>] [ENVIRONMENT IS HVTIP] [COPYING RECORDS INTO < COMMON LINKAGE WORKING > [COPYING DATA-NAMES INTO < COMMON LINKAGE WORKING > 	V	N			N
OVERLAY	<ul style="list-style-type: none"> Format: [OVERLAY <i>rec-delivery-area</i> WITH < ALL <i>rec-name-4</i> [,<i>rec-name-5</i>]>] [OVERLAY <i>rec-name-6</i> WITH <i>rec-name-7</i> [,<i>rec-name-8</i>]] 	V	N			N
POINTER	<ul style="list-style-type: none"> Format: [POINTER < AREAS > FOR INITIAL LOAD < IS ARE > <i>int-3</i>] 	V	N			N
PRIORITY	<ul style="list-style-type: none"> Format: [PRIORITY IS <i>int-1</i>] 	V	N			N
RECORD DELIVERY-AREA	<ul style="list-style-type: none"> Format: [RECORD DELIVERY-AREA IS <i>rec-delivery-area</i> [< WORKING COMMON LINKAGE >] [LENGTH IS <i>int-2</i> WORDS]] 	V	N			N

1-146 Supported COBOL Statements
Supported Unisys CDML COBOL statements

Clauses	Format	Parser	CE			
			Gen.	'Read' vari- ables	Written' vari- ables	DB
ROLLBACK	<ul style="list-style-type: none"> Format: [ROLLBACK IS <i>rollback-para</i>] 	V	N			N
RUN-UNIT-ID	<ul style="list-style-type: none"> Format: [RUN-UNIT-ID IS <i>run-unit-id</i>] 	V	N			N
SAVE DATA	<ul style="list-style-type: none"> Format: [SAVE DATA INCLUDES < < COMMAND RUN-UNIT > QUIT-BEFORE-LOOKS DEFFERED-UPDATES >]] 	VO	N			N

Procedure Division

Control Commands

Command	Format	Parser	CE			
			Gen.	Read variables	Written variables	DB
CLOSE	<ul style="list-style-type: none"> Format: CLOSE < ALL a-name-1 [,a-name-2] id-1 [,id-2 > [ON ERROR GO TO <i>error-para</i>]. 	V	S	a-name-1 a-name-2 id-1 id-2		N
DEPART	<ul style="list-style-type: none"> Format: DEPART [WITH < MESSAGE < ADVANCE TERMINATE > ROLLBACK [AND MESSAGE < DISCARD REQUEUE RETAIN >] >] [ON ERROR GO TO <i>error-para</i>]. 	V	S			N
IMPART	<ul style="list-style-type: none"> Format: IMPART [ON ERROR GO TO <i>error-para</i>]. 	V	S			N
OPEN	<ul style="list-style-type: none"> Format 1: OPEN ALL [USAGE-MODE IS < [<EXCLUSIVE PROTECTED>] < RETRIEVAL UPDATE > INITIAL LOAD >] . 	V	S			N
	<ul style="list-style-type: none"> Format 2: OPEN < a-name-1 id-1 > [USAGE-MODE IS < [<EXCLUSIVE PROTECTED>] < RETRIEVAL UPDATE > INITIAL LOAD >] < a-name-2 id-2 > [USAGE-MODE IS < [<EXCLUSIVE PROTECTED>] < RETRIEVAL UPDATE > INITIAL LOAD >]] [ON ERROR GO TO <i>error-para</i>] 	V	S	a-name-1 id-1 a-name-2 id-2		N

Retrieval Commands

Command	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ACQUIRE	<ul style="list-style-type: none"> Format ACQUIRE <int-1 id-1> DATABASE-KEYS [BEGINNING AT id-2 TO id-3 FROM <set-name-1 id-4> [USING DEFINED KEYS] [AT END GO TO <i>end-para</i>] [ON ERROR GO TO <i>error-para</i>] 	V	S	int-1 id-1 id-2 id-4	id-3	N

1-148 Supported COBOL Statements
Supported Unisys CDML COBOL statements

Command	Format	Parser	CE			
			Gen.	'Read' variables	Written' variables	DB
FIND/FETCH	<ul style="list-style-type: none"> Format < FIND FETCH > <i>rse</i> [SUPPRESS < ALL [< RECORD AREA < SETS <i>set-name-1</i> [,<i>set-name-2</i>] <i>id-1</i> [,<i>id-2</i>] >] > CURRENT RENCY UPDATES] [AT AND GO <i>end-para</i>] [ON ERROR GO TO <i>error-para</i>] 	V	S			N
rse as FIND statement argument	<ul style="list-style-type: none"> Format 1 [< <i>rec-name</i> <i>id-3</i> > RECORD] <i>id-4</i> [,<i>id-5</i>] 	V	S	<i>rec-name</i> <i>id-3</i> <i>id-4</i> <i>id-5</i>	<i>id-4</i> <i>id-5</i>	N
	<ul style="list-style-type: none"> Format 2 CURRENT RECORD WITHIN < <i>rec-name-2</i> <i>id-6</i> > RECORD 	V	S	<i>rec-name-2</i> <i>id-6</i>		N
	<ul style="list-style-type: none"> Format 3 < OWNER CURRENT NEXT PRIOR FIRST LAST <i>id-7</i> > RECORD WITHIN < <i>set-name-3</i> SET <i>id-8</i> SET <i>a-name-1</i> AREA <i>id-9</i> AREA > 	V	S	<i>id-7</i> <i>set-name-3</i> <i>id-8</i> <i>a-name-1</i> <i>id-9</i>		N
	<ul style="list-style-type: none"> Format 4 < NEXT PRIOR FIRST LAST <i>id-10</i> > < <i>rec-name-4</i> <i>id-11</i> > WITHIN < <i>set-name-4</i> SET <i>id-12</i> SET <i>a-name-2</i> AREA <i>id-13</i> AREA > 	V	S	<i>id-10</i> <i>rec-name-4</i> <i>id-11</i> <i>id-12</i> <i>a-name-2</i> <i>id-13</i>	<i>id-10</i>	N
	<ul style="list-style-type: none"> Format 5 [< NEXT [DUPLICATE] PRIOR FIRST LAST > WITHIN] < <i>rec-name-4</i> <i>id-14</i> > RECORD 	V	S	<i>rec-name-4</i> <i>id-14</i>		N
	<ul style="list-style-type: none"> Format 6 < <i>rec-name-5</i> <i>id-15</i> > VIA < <i>set-name-5</i> <i>id-16</i> > [USING <i>db-id-1</i> [,<i>db-id-2</i>]] 	V	S	<i>rec-name-5</i> <i>id-15</i> <i>set-name-5</i> <i>id-16</i> <i>db-id-1</i> <i>db-id-2</i>		N

Command	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
	<ul style="list-style-type: none"> Format 7 NEXT DUPLICATE [<i>rec-name-6</i> RECORD] WITHIN < <i>set-name-6</i> <i>id-17</i> > USING <i>db-id-3</i> [,<i>db-id-4</i>] 	V	S	<i>rec-name-6</i> <i>set-name-6</i> <i>id-17</i> <i>db-id-3</i> <i>db-id-4</i>		N
rse as FETCH statement argument	<ul style="list-style-type: none"> Format 1 [< <i>rec-name</i> <i>id-3</i> > RECORD] <i>id-4</i> [,<i>id-5</i>] 	V	S	<i>rec-name</i> <i>id-3</i> <i>id-4</i> <i>id-5</i>	<i>rec-name</i> <i>id-3</i> <i>id-4</i> <i>id-5</i>	N
	<ul style="list-style-type: none"> Format 2 CURRENT RECORD WITHIN < <i>rec-name-2</i> <i>id-6</i> > RECORD 	V	S	<i>rec-name-2</i> <i>id-6</i>	<i>rec-name-2</i> <i>id-6</i>	N
	<ul style="list-style-type: none"> Format 3 < OWNER CURRENT NEXT PRIOR FIRST LAST <i>id-7</i> > RECORD WITHIN < <i>set-name-3</i> SET <i>id-8</i> SET <i>a-name-1</i> AREA <i>id-9</i> AREA > 	V	S	<i>id-7</i> <i>set-name-3</i> <i>id-8</i> <i>a-name-1</i> <i>id-9</i>	<i>id-7</i>	N
	<ul style="list-style-type: none"> Format 4 < NEXT PRIOR FIRST LAST <i>id-10</i> > < <i>rec-name-4</i> <i>id-11</i> > WITHIN < <i>set-name-4</i> SET <i>id-12</i> SET <i>a-name-2</i> AREA <i>id-13</i> AREA > 	V	S	<i>id-10</i> <i>rec-name-4</i> <i>id-11</i> <i>id-12</i> <i>a-name-2</i> <i>id-13</i>	<i>id-10</i> <i>rec-name-4</i> <i>id-11</i>	N
	<ul style="list-style-type: none"> Format 5 [< NEXT [DUPLICATE] PRIOR FIRST LAST > WITHIN] < <i>rec-name-4</i> <i>id-14</i> > RECORD 	V	S	<i>rec-name-4</i> <i>id-14</i>	<i>rec-name-4</i> <i>id-14</i>	N
	<ul style="list-style-type: none"> Format 6 < <i>rec-name-5</i> <i>id-15</i> > VIA < <i>set-name-5</i> <i>id-16</i> > [USING <i>db-id-1</i> [,<i>db-id-2</i>]] 	V	S	<i>rec-name-5</i> <i>id-15</i> <i>set-name-5</i> <i>id-16</i> <i>db-id-1</i> <i>db-id-2</i>	<i>rec-name-5</i> <i>id-15</i>	N

1-150 Supported COBOL Statements
Supported Unisys CDML COBOL statements

Command	Format	Parser	CE			
			Gen.	'Read' variables	Written' variables	DB
	<ul style="list-style-type: none"> Format 7 NEXT DUPLICATE [<i>rec-name-6</i> RECORD] WITHIN < <i>set-name-6</i> <i>id-17</i> > USING <i>db-id-3</i> [, <i>db-id-4</i>] 	V	S	<i>rec-name-6</i> <i>set-name-6</i> <i>id-17</i> <i>db-id-3</i> <i>db-id-4</i>		N
GET	<ul style="list-style-type: none"> Format GET [< <i>rec-name</i> <i>id-1</i> > RECORD] [ON ERROR GO TO <i>error-para</i>] 	V	S		<i>id-1</i> <i>rec-name</i>	N

Update Commands

Command	Format	Parser	CE			
			Gen.	'Read' variables	Written' variables	DB
DELETE	<ul style="list-style-type: none"> Format DELETE [< <i>rec-name</i> <i>id-1</i> > RECORD] [< ONLY ALL >] [ON ERROR GO TO <i>error-para</i>] 	V	S	<i>rec-name</i> <i>id-1</i>		N
INSERT	<ul style="list-style-type: none"> Format 1 INSERT [< <i>rec-name</i> <i>id-1</i> > RECORD] INTO < <i>set-name -1</i> <i>id-2</i> > [, INTO < <i>set-name-2</i> <i>id-3</i> >] [ON ERROR GO TO <i>error-para</i>] 	V	S	<i>id-1</i> <i>rec-name</i>	<i>set-name-1</i> <i>id-2</i> <i>set-name-2</i> <i>id-3</i>	N
	<ul style="list-style-type: none"> Format 2 INSERT [< <i>rec-name</i> <i>id-1</i> > RECORD] INTO ALL SETS [ON ERROR GO TO <i>error-para</i>] 	V	S	<i>rec-name</i> <i>id-1</i>		N
MODIFY	<ul style="list-style-type: none"> Format MODIFY [< <i>rec-name</i> <i>id-1</i> > RECORD] [<i>db-id-1</i> [, <i>db-id-2</i>]] [ON ERROR GO TO <i>error-para</i>] 	V	S	<i>rec-name</i> <i>id-1</i> <i>db-id-1</i> <i>db-id-2</i>		N

Command	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
REMOVE	<ul style="list-style-type: none"> Format 1 REMOVE [< rec-name id-1 > RECORD] FROM < set-name -1 id-2 > [, < set-name-2 id-3 >] [ON ERROR GO TO error-para] 	V	S	rec-name id-1 set-name-1 id-2 set-name-2 id-3		N
	<ul style="list-style-type: none"> Format 2 REMOVE [< rec-name id-1 > RECORD] FROM ALL SETS [ON ERROR GO TO error-para] 	V	S	rec-name id-1		N
STORE	<ul style="list-style-type: none"> Format STORE < rec-name id-1 > [SUPPRESS < ALL [< RECORD AREA < SETS set-name-1 [,set-name-2] id-2 [,id-3] >] > CURRENCY UPDATES] [ON ERROR GO TO error-para] 	V	S	rec-name-1 id-1		N

Support Commands

Command	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
FREE	<ul style="list-style-type: none"> Format FREE [WITH MESSAGE < ADVANCE TERMINATE >] [ON ERROR GO TO error-para] 	V	S			N
IF	<ul style="list-style-type: none"> Format 1 IF < set-name-1 id-1 > SET [NOT] EMPTY; < statement-1 NEXT SENTENCE > [ELSE < statement-2 NEXT SENTENCE >] 	V	S	set-name-1 id-1		N
	<ul style="list-style-type: none"> Format 2 IF RECORD [NOT] < OWNER MEMBER > OF < set-name-1 id-1 ANY > SET < statement-3 NEXT SENTENCE > [ELSE < statement-4 NEXT SENTENCE >] 	V	S	set-name-1 id-1		N

1-152 Supported COBOL Statements
Supported Unisys CDML COBOL statements

Command	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
KEEP	<ul style="list-style-type: none"> Format KEEP [< rec-name id-1> RECORD] [ON ERROR GO TO <i>error-para</i>] 	V	S	rec-name id-1		N
LOG	<ul style="list-style-type: none"> Format KEEP id-1 WORDS [FROM id-2] [FOR RECOVERY POINT] [ON ERROR GO TO <i>error-para</i>] 	V	S	id-1 id-2		N
MOVE	<ul style="list-style-type: none"> Format 1 MOVE CURRENCY STATUS FOR < RUN-UNIT <i>set-name</i> SET id-3 SET <i>a-name</i> AREA id-2 AREA id-1 RECORD <i>rec-name</i> RECORD > TO id-4 [ON ERROR GO TO <i>error-para</i>] 	V	S	set-name id-3 a-name id-2 id-1 rec-name	id-4	N
	<ul style="list-style-type: none"> Format 2 MOVE < AREA-KEY AREA-NAME > FOR < RUN-UNIT <i>set-name</i> SET id-7 SET <i>a-name</i> AREA id-6 AREA id-5 RECORD <i>rec-name</i> RECORD id-8 > TO < id-4 id-10 > [ON ERROR GO TO <i>error-para</i>] 	V	S	set-name id-7 a-name id-6 id-5 rec-name id-8	id-4 id-10	N

Supported MicroFocus COBOL statements

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S
NULL / NULLS	V	S

Special registers

Registers	Parser	CE
ADDRESS OF	V	PS
LENGTH OF	V	S
RETURN-CODE	V	N
SORT-CONTROL	V	N
SORT-RETURN	V	N

1-154 Supported COBOL Statements
Supported MicroFocus COBOL statements

Literals

Literal type	Format	Parser	CE
Nonnumeric literals	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
	<ul style="list-style-type: none"> Format 2: with apostrophes Example: 'THIS ISN'T WRONG' 	V	S
	<ul style="list-style-type: none"> Format 3: hexadecimal notation <i>X"hexadecimal-digits"</i> 	V	S
Numeric literals	<ul style="list-style-type: none"> Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
	<ul style="list-style-type: none"> Format 3: hexanumeric <i>X"hexadecimal-digits"</i> 	V	S
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format <i>text-name-1 [<IN OF> library-name-1]</i> 	V	S
to Procedure Division	<ul style="list-style-type: none"> Format 1: <i>paragraph-name-1 [<IN OF> section-name-1]</i> 	V	S
	<ul style="list-style-type: none"> Format 2: <i>section-name-1</i> 	V	S

Division	Format	Parser	CE
to Data Division	<ul style="list-style-type: none"> Format 1: simple data reference <i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> Format 2: subscripting <i>data-name-1</i> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) (<i>leftmost-character-position: [length]</i>) 	V	S
	<ul style="list-style-type: none"> Format 3: <<i>condition-name-1</i> <i>data-name-1</i>> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 4: LINAGE-COUNTER [<IN OF> <i>file-name-2</i>] 	V	S
Condition names	<ul style="list-style-type: none"> Format 1: Data Division <i>condition-name-1</i> [<IN OF> <i>data-name-1</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) 	V	S
	<ul style="list-style-type: none"> Format 2: Special-Names paragraph <i>condition-name-1</i> [<IN OF> <i>mnemonic-name-1</i>] 	V	S
Subscript	<ul style="list-style-type: none"> Format <<i>integer-1</i> ALL <i>data-name-3</i> [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>> 	V	S
Reference modification	<ul style="list-style-type: none"> Format <<i>data-name-1</i> FUNCTION <i>function-name-1</i> (<i>arguments</i>)> (<i>leftmost-character-position: [length]</i>) 	V	S

Identification Division

Paragraphs	Format	Parser	CE
PROGRAM-ID paragraph	<ul style="list-style-type: none"> Format [<ID IDENTIFICATION> DIVISION] [PROGRAM-ID. <i>program-name</i> [< IS <COMMON INITIAL> PROGRAM IS EXTERNAL PROGRAM >]]. [AUTHOR. [<i>comment-entry</i>]] [INSTALLATION. [<i>comment-entry</i>]] [DATE-WRITTEN. [<i>comment-entry</i>]] [DATE-COMPILED. [<i>comment-entry</i>]] [SECURITY. [<i>comment-entry</i>]] 	V	S

Environment Division

Input-Output Section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential file-control entries SELECT [<OPTIONAL NOT OPTIONAL>] <i>file-name-1</i> ASSIGN TO < [< EXTERNAL DYNAMIC>] [<< LINE ADVANCING MULTIPLE < REEL UNIT >> FILE DISK >] < <i>external-file-reference</i> <i>data-name-1</i> <i>literal-1</i>> [< EXTERNAL DYNAMIC>] < DISK KEYBOARD DISPLAY PRINTER PRINTER-1> [< <i>external-file-reference</i> <i>data-name-1</i> <i>literal-1</i>>] DISK FROM <i>data-name-1</i>> 	V	S
	[RESERVE <i>integer-1</i> < AREA AREAS >] [[ORGANISATION IS] [<RECORD>] SEQUENTIAL] [PADDING [CHARACTER] [IS] < <i>data-name-5</i> <i>literal-2</i> >] [RECORD DELIMITER [IS] <STANDARD-1 character-string>] [ACCESS MODE IS SEQUENTIAL] [LOCK MODE IS << MANUAL AUTOMATIC> [WITH <LOCK ON [MULTIPLE] < RECORD RECORDS> ROLLBACK>] EXCLUSIVE>]	VO	N
	<ul style="list-style-type: none"> Format 2: indexed file-control entries SELECT [< OPTIONAL NOT OPTIONAL>] <i>file-name-1</i> ASSIGN TO < [< EXTERNAL DYNAMIC>] DISK < <i>external-file-reference</i> <i>literal-1</i>> [< EXTERNAL DYNAMIC>] DISK FROM <i>data-name-1</i>> 	V	S
	[RESERVE <i>integer-1</i> < AREA AREAS >] [ORGANIZATION IS] INDEXED [ACCESS MODE IS <SEQUENTIAL RANDOM DYNAMIC>] [LOCK MODE IS << MANUAL AUTOMATIC> [WITH <LOCK ON [MULTIPLE] < RECORD RECORDS> ROLLBACK>] EXCLUSIVE>] RECORD KEY IS < <i>data-name-5</i> <i>split-key-name-1</i> = <i>data-name-6</i> [<i>data-name-7</i>]> [ALTERNATE [RECORD] KEY IS < <i>data-name-8</i> <i>split-key-name-2</i> = <i>data-name-9</i> [<i>data-name-10</i>]> [WITH DUPLICATES] [SUPPRESS [WHEN] < ZERO S ZEROES SPACES [ALL] <i>literal</i> >]] [FILE STATUS IS <i>data-name-2</i>]	VO	N

Paragraphs and entries	Format	Parser	CE
	<ul style="list-style-type: none"> Format 3: relative file-control entries SELECT [< OPTIONAL NOT OPTIONAL>] <i>file-name-1</i> ASSIGN TO < [< EXTERNAL DYNAMIC>] DISK < <i>external-file-reference</i> <i>literal-1</i> > [< EXTERNAL DYNAMIC>] DISK FROM <i>data-name-1</i>> 	V	S
	[RESERVE <i>integer-1</i> < AREA AREAS >] [ORGANIZATION IS] RELATIVE [ACCESS MODE IS <SEQUENTIAL [RELATIVE KEY IS <i>data-name-5</i>] <RANDOM DYNAMIC> RELATIVE KEY IS <i>data-name-5</i> >] [LOCK MODE IS << MANUAL AUTOMATIC> [WITH <LOCK ON [MULTIPLE] < RECORD RECORDS> ROLLBACK>] EXCLUSIVE>] [FILE STATUS IS <i>data-name-2</i>]	VO	N
	<ul style="list-style-type: none"> Format 4: line-sequential file-control entries SELECT [<OPTIONAL NOT OPTIONAL>] <i>file-name-1</i> ASSIGN TO < [< EXTERNAL DYNAMIC>] [<FILE DISK >] < <i>external-file-reference</i> <i>data-name-1</i> <i>literal-1</i> > [< EXTERNAL DYNAMIC>] < DISK KEYBOARD DISPLAY PRINTER PRINTER-1> [< <i>external-file-reference</i> <i>data-name-1</i> <i>literal-1</i> >] DISK FROM <i>data-name-1</i>> 	V	S
	[[ORGANISATION IS] [LINE] SEQUENTIAL] [ACCESS MODE IS SEQUENTIAL] [LOCK MODE IS << MANUAL AUTOMATIC> [WITH <LOCK ON [MULTIPLE] < RECORD RECORDS> ROLLBACK>] EXCLUSIVE>] [FILE STATUS IS <i>data-name-2</i>] [PASSWORD IS <i>data-name-4</i>]	VO	N
	<ul style="list-style-type: none"> Format 5: sort-merge file-control entries SELECT <i>file-name</i> ASSIGN TO < <i>external-file-reference</i> <i>data-name-1</i> <i>literal-1</i> > [<SORT FILE> STATUS IS <i>data-name-2</i>] 	V	S
	SELECT <i>file-name</i> ASSIGN TO <EXTERNAL DYNAMIC > < <i>external-file-reference</i> <i>data-name-1</i> <i>literal-1</i> > [<SORT FILE> STATUS IS <i>data-name-2</i>]	VO	N

1-158 Supported COBOL Statements
Supported MicroFocus COBOL statements

Configuration Section

Paragraphs and entries	Format	Parser	CE
CONFIGURATION section	<ul style="list-style-type: none"> • Format [CONFIGURATION SECTION] [SOURCE-COMPUTER. [<i>source-computer-name</i> [WITH DEBUGGING MODE].]] [OBJECT-COMPUTER. <i>object-computer-name</i> [MEMORY SIZE <i>integer</i> <WORDS CHARACTERS MODULES>] [PROGRAM COLLATING SEQUENCE IS <i>alphabet-name</i>] [SEGMENT-LIMIT IS <i>segment-number</i>.] [SPECIAL-NAMES. [< SWITCH-0 SWITCH-1 SWITCH-2 SWITCH-3 SWITCH-4 SWITCH-5 SWITCH-6 SWITCH-7 SWITCH-8 <i>function-name</i> > <[IS] <i>mnemonic-name</i> [ON STATUS [IS] <i>condition-name-1</i> [OFF STATUS] [IS] <i>condition-name-2</i>] [IS] <i>mnemonic-name</i> [OFF STATUS [IS] <i>condition-name-2</i> [ON STATUS] [IS] <i>condition-name-1</i>] [ON STATUS [IS] <i>condition-name-1</i> [OFF STATUS [IS] <i>condition-name-2</i>] [OFF STATUS [IS] <i>condition-name-2</i> [ON STATUS [IS] <i>condition-name-1</i>]]>] [ALPHABET <i>alphabet-name-1</i> IS <<STANDARD-1 STANDARD-2 NATIVE ASCII EBCDIC> <<i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i>>>] [SYMBOLIC CHARACTERS <<<i>symbolic-character</i> <ARE IS> <<i>integer-1</i>>> [IN <i>alphabet-name-2</i>]] [CLASS <i>class-name-1</i> IS <<i>literal-4</i> [<THROUGH THRU> <i>literal-5</i>>] [CURRENCY SIGN IS <i>literal-6</i>] [DECIMAL-POINT [IS] COMMA] [NUMERIC SIGN IS TRAILING SEPARATE] [CALL-CONVENTION <i>integer-2</i> IS <i>mnemonic-name</i>] [CONSOLE IS CRT] [INPUT-OUTPUT SECTION.] [FILE-CONTROL.] <<i>file-control-entry</i>> [I-O-CONTROL. [RERUN [ON< <i>file-name-1</i> <i>character-string</i>>] EVERY << [END OF] < REEL UNIT > <i>integer-1</i> RECORDS> OF <i>file-name-2</i> <i>integer-2</i> CLOCK-UNITS <i>condition-name</i> >] [SAME [< RECORD SORT SORT-MERGE] AREA FOR <i>file-name-3</i> <<i>file-name-4</i>>] [MULTIPLE FILE TAPE CONTAINS <<i>file-name-5</i> [POSITION <i>integer-3</i>]>] 	VO	N

Data Division

Entries and Clauses	Format	Parser	CE
Data Division paragraph	<ul style="list-style-type: none"> Format [DATA DIVISION.] [[FILE SECTION.] [< file-description-entry < record-description-entry > >sort-merge-file-description-entry <record-description-entry> report-file-description-entry >]] [WORKING-STORAGE SECTION. [77-level-description-entry record-description-entry]] [LOCAL-STORAGE SECTION. [77-level-description-entry record-description-entry]] [LINKAGE SECTION. [77-level-description-entry record-description-entry]] [REPORT SECTION [report-description-entry < report-group-description-entry>]] [SCREEN SECTION [screen-description-entry]]] [COMMUNICATION SECTION. [communication-description-entry [record-description-entry]]] 	V	S
File Description (FD) paragraph	<ul style="list-style-type: none"> Format 1: sequential files FD file-name-1 <[IS EXTERNAL] [IS GLOBAL]> [BLOCK CONTAINS [integer-1 TO] integer-2 <CHARACTERS RECORDS>] [RECORD <CONTAINS integer-3 CHARACTERS CONTAINS integer-6 TO integer-7 CHARACTERS IS VARYING IN SIZE [[FROM] integer-4] [TO integer-5] CHARACTERS] [DEPENDING ON data-name-1]> [LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED>] [VALUE OF <implementor-name-1 IS <data-name-3 literal-1> FILE-ID IS < data-name-6 literal-3 >>] [RECORDINFG MODE IS < F V U S FIXED VARIABLE >] [DATA <RECORD IS RECORDS ARE> data-name-7 [data-name-8]] [LINAGE IS <data-name-9 integer-8> LINES [WITH FOOTING AT <data-name-10 integer-9>] [LINES AT TOP <data-name-11 integer-10>] [LINES AT BOTTOM <data-name-12 integer-11>] [CODE-SET IS alphabet-name [FOR <id-1>]]. <record-description-entry> Format 2: relative/indexed files FD file-name-1 <[IS EXTERNAL] [IS GLOBAL]> [BLOCK CONTAINS [integer-1 TO] integer-2 <CHARACTERS RECORDS>] [RECORD <CONTAINS integer-3 CHARACTERS CONTAINS [integer-6 TO] integer-7 CHARACTERS IS VARYING IN SIZE [[FROM integer-4] [TO integer-5] CHARACTERS] [DEPENDING [ON] data-name-1]>] [LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED>] [VALUE OF <implementor-name-1 IS <data-name-3 literal-1> FILE-ID IS <data-name-6 literal-3>>] [RECORDINFG MODE IS < F V U S FIXED VARIABLE >] [DATA <RECORD IS RECORDS ARE> data-name-7 [data-name-8]] < record-description-entry > 	V VO	S N

1-160 Supported COBOL Statements
Supported MicroFocus COBOL statements

Entries and Clauses	Format	Parser	CE
	<ul style="list-style-type: none"> Format 3: line-sequential files FD <i>file-name-1</i> 	V	S
	<p><[IS EXTERNAL] [IS GLOBAL]> [BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <CONTAINS <i>integer-3</i> CHARACTERS IS VARYING IN SIZE [FROM <i>integer-4</i>] [TO <i>integer-5</i>] CHARACTERS [DEPENDING ON <i>data-name-1</i>] CONTAINS [<i>integer-6</i> TO] <i>integer-7</i> CHARACTERS >] [LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED>] [VALUE OF <<i>implementor-name-1</i> IS <<i>data-name-3</i> <i>literal-1</i>> FILE-ID IS < <i>data-name-6</i> <i>literal-3</i> >>] [RECORDING MODE IS < F V FIXED VARIABLE >] [DATA <RECORD IS RECORDS ARE> <i>data-name-7</i> [<i>data-name-8</i>]] [CODE-SET IS <i>alphabet-name</i> [FOR <<i>id-1</i>>]]. <<i>record-description-entry</i>></p>	VO	N
	<ul style="list-style-type: none"> Format 4: sort/merge files SD <i>file-name-1</i> 	V	S
	<p>[RECORD <CONTAINS <i>integer-3</i> CHARACTERS CONTAINS [<i>integer-4</i> TO] <i>integer-5</i> CHARACTERS IS VARYING IN SIZE [[FROM <i>integer-6</i>] [TO <i>integer-7</i>] CHARACTERS] [DEPENDING ON <i>data-name-1</i>]]> [DATA <RECORD IS RECORDS ARE> <<i>data-name-3</i>>] [VALUE OF FILE-ID IS <<i>data-name-4</i> <i>literal-1</i>>]</p>	VO	N
	<ul style="list-style-type: none"> Format 5: report files FD <i>file-name-1</i> 	V	S
	<p><[IS EXTERNAL] [IS GLOBAL]FD <i>file-name-1</i> <[IS EXTERNAL] [IS GLOBAL]> [BLOCK CONTAINS [<i>integer-1</i> TO] <i>integer-2</i> <CHARACTERS RECORDS>] [RECORD <CONTAINS <i>integer-3</i> CHARACTERS CONTAINS <i>integer-6</i> TO <i>integer-7</i> CHARACTERS IS VARYING IN SIZE [[FROM] <i>integer-4</i>] [TO <i>integer-5</i>] CHARACTERS] [DEPENDING ON <i>data-name-1</i>]]> [LABEL <RECORD IS RECORDS ARE> <STANDARD OMITTED>] [VALUE OF <<<i>data-name-3</i> IS <<i>data-name-4</i> <i>literal-1</i>>> FILE-ID IS < <i>data-name-3</i> <i>literal-1</i> >>] [RECORDINFG MODE IS < F V FIXED VARIABLE >] [DATA <RECORD IS RECORDS ARE> <i>data-name-4</i> [<i>data-name-5</i>]] [CODE-SET IS <i>alphabetic-name-1</i>]. [< REPORT IS REPORT ARE > <<i>report-name-1</i>>]</p>	VO	N

Entries and Clauses	Format	Parser	CE
Data Description Entry	<ul style="list-style-type: none"> • Format 1 <i>level number</i> [< <i>data-name-1</i> FILLER >] [REDEFINES <i>data-name-2</i>] [IS GLOBAL] [IS TYPEDEF] [< PICTURE PIC > IS <i>character-string</i>] [[USAGE IS] < BINARY COMPUTATIONAL COMP COMPUTATIONAL-1 COMP-1 COMPUTATIONAL-2 COMP-2 COMPUTATIONAL-3 COMP-3 COMPUTATIONAL-4 COMP-4 COMPUTATIONAL-5 COMP-5 COMPUTATIONAL-X COMP-X <i>typedef-name-1</i> INDEX PACKED-DECIMAL DISPLAY POINTER DISPLAY-1 PROCEDURE-POINTER >] [< OCCURS <i>integer-2</i> TIMES [< ASCENDING DESVENDING > KEY IS <<i>data-name-3</i>>] [INDEXED BY <<i>index-name-1</i>>] OCCURS [<i>integer-1 TO</i>] <i>integer-2</i> TIMES DEPENDING ON <i>data-name-4</i> [< ASCENDING DESCENDING > KEY IS <<i>data-name-3</i>>] [INDEXED BY <<i>index-name-1</i>>] >] [SIGN IS < LEADING TRAILING > [SEPARATE CHARACTER]] [< SYNCHRONIZED SYNC > [<LEFT RIGHT>]] [< JUSTIFIED JUST > RIGHT] [BLANK WHEN < ZERO ZEROS ZEROES >] [VALUE IS <i>literal-1</i>] 	V	S
	<ul style="list-style-type: none"> • Format 2 66 <i>data-name-1</i> RENAMES <i>data-name-2</i> [< THROUGH THRU > <i>data-name-3</i>]. 	V	S
	<ul style="list-style-type: none"> • Format 3 88 <i>condition-name</i> < VALUE IS VALUE ARE > [<i>literal-2</i> < THROUGH THRU > <i>literal-3</i>] [WHEN SET TO FALSE <i>literal-4</i>]. 	V	S
	<ul style="list-style-type: none"> • Format 4 78 <i>constant-name</i> VALUE IS < <i>literal-5</i> NEXT START OF <i>data-name-1</i> LENGHT OF <i>data-name-2</i> > [< + - * / AND OR > < <i>integer-1</i> NEXT START OF <i>data-name-3</i> LENGHT OF <i>data-name-4</i> >]. 	V	S

1-162 Supported COBOL Statements
Supported MicroFocus COBOL statements

Entries and Clauses	Format	Parser	CE
Communication Description Entry	<ul style="list-style-type: none"> Format 1 CD <i>cd-name</i> FOR [INITIAL] INPUT [< [SYMBOLIC QUEUE IS <i>data-name-1</i>] < [SYMBOLIC SUB-QUEUE IS <i>data-name-2</i>] [SYMBOLIC SUB-QUEUE IS <i>data-name-3</i>] [SYMBOLIC SUB-QUEUE IS <i>data-name-4</i>] [MESSAGE DATE IS <i>data-name-5</i>] [MESSAGE TIME IS <i>data-name-6</i>] [SYMBOLIC SOURCE IS <i>data-name-7</i>] [TEXT LENGHT IS <i>data-name-8</i>] [END KEY IS <i>data-name-9</i>] [STATUS KEY IS <i>data-name-10</i>] [MESSAGE COUNT IS <i>data-name-11</i>] >] [<i>data-name-1 data-name-2...data-name-11</i>] >] 	VO	N
	<ul style="list-style-type: none"> Format 2 CD <i>cd-name</i> FOR OUTPUT [DESTINATION COUNT IS <i>data-name-1</i>] [TEXT LENGHT IS <i>data-name-2</i>] [STATUS KEY IS <i>data-name-3</i>] [DESTINATION TABLE OCCURS <i>integer-2</i> TIMES [INDEXED BY <i>index-name-1</i> [<i>index-name-2</i>]] 	VO	N
	<ul style="list-style-type: none"> Format 3 CD <i>cd-name</i> FOR [INITIAL] I-O [<< [MESSAGE DATE IS <i>data-name-1</i>] [MESSAGE TIME IS <i>data-name-2</i>] [SYMBOLIC TERMINAL IS <i>data-name-3</i>] [TEXT LENGHT IS <i>data-name-4</i>] [END KEY IS <i>data-name-5</i>] [STATUS KEY IS <i>data-name-6</i>] > [<i>data-name-1 data-name-2...data-name-6</i>] >] 	VO	N
Report Description Entry	RD <i>report-name-1</i> [IS GLOBAL] [CODE < <i>literal-1</i> >] [< CONTROL IS CONTROL ARE > < <i>data-name-1</i> > FINAL [<i>data-name-1</i>] >] [PAGE [< LIMIT IS LIMITS ARE >] <i>integer-1</i> [< LINE LINES >] [HEADING <i>integer-2</i>] [FIRST DETAIL <i>integer-3</i>] [LAST DETAIL <i>integer-4</i>] [FOOTING <i>integer-5</i>]	VO	N

Entries and Clauses	Format	Parser	CE
Report Group Description Entry	<ul style="list-style-type: none"> Format 1 01 [<i>data-name-1</i>] 	V	S
	<p>[LINE NUMBER IS < <i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i> >] [NEXT GROUP IS < <i>integer-3</i> PLUS <i>integer-4</i> NEXT PAGE >] TYPE IS < < REPORT HEADING RH > < PAGE HEADING PH > < CONTROL HEADING CH > < <i>data-name-2</i> FINAL > < DETAIL DE > < CONTROL FOOTING CF > < <i>data-name-3</i> FINAL > < PAGE FOOTING PF > < REPORT FOOTING RF > > [[USAGE IS] < DISPLAY DISPLAY-1 >.</p>	VO	N
	<ul style="list-style-type: none"> Format 2 <i>level-number</i> [<i>data-name-1</i>] 	V	S
	<p>[LINE NUMBER IS < <i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i> >] [[USAGE IS] < DISPLAY DISPLAY-1 >.</p>	VO	N
	<ul style="list-style-type: none"> Format 3 <i>level-number</i> [<i>data-name-1</i>] <p>< PICTURE PIC > IS <i>character-string</i> [[USAGE IS] < DISPLAY DISPLAY-1 > [SIGN IS < LEADING TRAILING > [SEPARATE CHARACTER]] [< JUSTIFIED JUST > RIGHT] [BLANK WHEN < ZERO ZEROS ZEROES >] [LINE NUMBER IS < <i>integer-1</i> [ON NEXT PAGE] PLUS <i>integer-2</i> >] [COLUMN NUMBER IS <i>integer-3</i>] < SOURCE IS <i>id-1</i> VALUE IS <i>literal-1</i> < SUM <<i>id-2</i>>> [UPON <<i>data-name-2</i>>] [RESET ON < <i>data-name-3</i> FINAL>] > [GROUP INDICATE]</p>	VO	N

1-164 Supported COBOL Statements
Supported MicroFocus COBOL statements

Entries and Clauses	Format	Parser	CE
Screen Description Entry	<pre> level-number [screen-name FILLER] [BLANK < SCREEN LINE >] [< BELL BEEP >] [BLINK] [ERASE < EOL EOS >] [< HIGHLIGHT LOWLIGHT >] [GRID] [LEFTLINE] [OVERLINE] [REVERSE-VIDEO] [UNDERLINE] [SIZE IS < id-1 integer-1 >] [LINE [NUMBER IS [< PLUS + - >] < id-2 integer-2>]] [< COLUMN COL > [NUMBER IS [< PLUS + - >] < id-3 integer-3>]] [< FOREGROUND-COLOR FOREGROUND-COLOUR > IS < integer-4 id-4 >] [< BACKGROUND-COLOR BACKGROUND-COLOUR > IS < integer-5 id-5 >] [CONTROL IS id-6] [[VALUE IS] literal-6] [< PICTURE PIC > IS character-string] [< [FROM < id-6 literal-2>] [TO id-7] USING id-8 >] [[USAGE IS] < DISPLAY DISPLAY-1 >] [BLANK WHEN ZERO] [< JUSTIFIED JUST > RIGHT] [[SIGN IS] < LEADING TRAILING > [SEPARATE CHARACTER]] [< AUTO AUTO-SKIP >] [< SECURE NO-ECHO >] [< REQUIRED EMPTY-CHECK >] [PROMPT [CHARACTER IS < id-9 literal-3 >]] [OCCURS integer-6 TIMES] [< FULL LENGHT-CHECK >] [ZERO-FILL] </pre>	VO	N

Procedure Division

Entries and Clauses	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
Procedure Division paragraph	<ul style="list-style-type: none"> Declarative Format <i>procedure-division-header</i> [DECLARATIVES. < <i>section-name</i> SECTION [segment-number] [<i>declarative-sentence</i>] [<i>paragraph-name</i> [<i>sentence</i>]] > END DECLARATIVES.] [[<i>sentence</i>] [<i>paragraph-name</i> [<i>sentence</i>]] < <i>section-name</i> SECTION [segment-number] [<i>sentence</i>] [<i>paragraph-name</i> [<i>sentence</i>]] > 	V	S			N
	<ul style="list-style-type: none"> Non-Declarative Format <i>[procedure-division-header]</i> [<i>sentence</i>] [<i>paragraph-name</i> < <i>sentence</i> >] 	V	S			N

Conditional Expressions

Condition	Format	Parser	CE			
			Gen.	'Read' Variables	'Written' Variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1</i> IS [NOT] <NUMERIC ALPHABETIC ALPHABETIC-LOWER ALPHABETIC-UPPER <i>class-name-1</i> > 	V	S	id-1		S
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	condition-name		S
Relation condition	<ul style="list-style-type: none"> Format < <i>id-1</i> <i>literal-1</i> <i>arithmetic-expression-1</i> <i>index-name-1</i> > IS < [NOT] < GREATER THAN LESS THAN EQUAL TO < > = > GREATER THAN EQUAL TO >= LESS THAN OR EQUAL TO <= > IS UNEQUAL TO IS <> EQUALS EXCEEDS > < <i>id-2</i> <i>literal-2</i> <i>arithmetic-expression-2</i> <i>index-name-2</i> > 	V	S	id-1 index-name-1 id-2 arithmetic-expression-2 index-name-2		S

1-166 Supported COBOL Statements
Supported MicroFocus COBOL statements

Condition	Format	Parser	CE			
			Gen.	'Read' Variables	'Written' Variables	DB
Sign condition	<ul style="list-style-type: none"> Format operand-1 [IS] [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	operand-1		S
Switch-status condition	<ul style="list-style-type: none"> Format condition-name 	V	S	condition-name		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT simple-condition-1 	V	S	simple-condition-1		S
Combined conditions	<ul style="list-style-type: none"> Format condition-1 <<AND OR> condition-1> 	V	S	condition-1 condition-2		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format relation-condition <<AND OR> [NOT] [relational-operator] object> 	V	S	relation-condition object		S
Pointer condition	<ul style="list-style-type: none"> Format < ADDRESS OF id-1 id-2 NULL > < IS [NOT] EQUAL TO IS [NOT] = IS UNEQUAL TO IS <> EQUALS > < ADDRESS OF id-3 id-4 NULL > 	V	S	id-1 id-2 id-3 id-4		N
Procedure-Pointer Condition	<ul style="list-style-type: none"> Format < id-1 NULL > < IS [NOT] EQUAL TO IS [NOT] = IS UNEQUAL TO IS <> EQUALS > < id-2 NULL > 	N	N			N
Object Relation Condition	<ul style="list-style-type: none"> Format < object-id-1 NULL > < IS [NOT] EQUAL TO IS [NOT] = IS UNEQUAL TO IS <> EQUALS > < object-id-2 NULL > 	N	N			N

Statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format 1 PROCEDURE DIVISION [<i>mnemonic-name</i>] [< USING CHAINING > < [BY REFERENCE] < <i>data-name-1</i> > BY VALUE <<i>data-name-2</i>> >] 	V	S	<i>data-name-1</i>	<i>data-name-1</i> <i>data-name-2</i>	N
	<ul style="list-style-type: none"> Format 2 PROCEDURE DIVISION [<i>mnemonic-name</i>] [< USING CHAINING > < [BY REFERENCE] < <i>data-name-1</i> [DELIMITED [BY SIZE]] typedef-name-1 ANY > BY VALUE < <i>data-name-2</i> typedef-name-2 ANY > > [REPEATED [<i>integer-1</i> TO <i>integer-2</i>]]] [< GIVING RETURNING > < <i>data-name-3</i> typedef-name-3 >] 	N	N			N
ACCEPT	<ul style="list-style-type: none"> Format 1 ACCEPT <i>id</i> [FROM <<i>mnemonic-name</i> <i>function-name</i>>] < [ON EXCEPTION <i>imperative-statement-1</i>] [NOT ON EXCEPTION <i>imperative-statement-2</i>] [END-ACCEPT] > 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 2 ACCEPT <i>id</i> FROM < DATE DAY DAY-OF-WEEK TIME> [END-ACCEPT] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 3 ACCEPT <i>id</i> FROM < LINE NUMBER USER NAME ESCAPE KEY EXCEPTION STATUS > [END-ACCEPT] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 4 ACCEPT <i>screen-name</i> [AT < <i>id-4</i> <i>Integer-3</i> >] [ON < EXCEPTION ESCAPE > <i>imperative-statement-1</i>] [NOT ON < EXCEPTION ESCAPE > <i>imperative-statement-2</i>] [END-ACCEPT] 	V	S	<i>id-4</i>	<i>screen-name</i>	N

1-168 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
ACCEPT	<ul style="list-style-type: none"> Format 5 ACCEPT <i>id-1</i> [< AT [LINE NUMBER < <i>id-2</i> <i>integer-1</i> >] [< COLUMN COL > NUMBER < <i>id-3</i> <i>integer-2</i> >] AT < <i>id-4</i> <i>integer-3</i> > >] [FROM CRT] [MODE IS BLOCK] [WITH < < AUTO AUTO-SKIP > < BELL BEEP > BLINK < FULL LENGHT-CHECK > GRID [< HIGHLIGHT LOW-LIGHT >] < LEFTLINE OVERLINE PROMPT [CHARACTER IS [<i>literal-1</i>]] > < REQUIRED EMPTY-CHECK > REVERSE-VIDEO < SECURE NO-ECHO > SIZE IS < <i>id-6</i> <i>integer-4</i> > UNDERLINE < FOREGROUND-COLOR FOREGROUND-COLOUR > IS <i>integer-5</i> < BACKGROUND-COLOR BACKGROUND-COLOUR > IS <i>integer-6</i> CONTROL IS < <i>id-7</i> <i>literal-2</i> > < TIME-OUT TIMEOUT > AFTER < <i>integer-7</i> <i>id-8</i> > < LEFT-JUSTIFY RIGHT-JUSTIFY SPACE-FILL TRAILING-SIGN UPDATE UPPER LOWER ZERO-FILL > >] [ON < EXCEPTION ESCAPE > <i>imperative-statement-1</i>] [NOT ON < EXCEPTION ESCAPE > <i>imperative-statement-2</i>] [END-ACCEPT] 	V	S	<i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-6</i> <i>id-7</i> <i>id-8</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 6 ACCEPT <i>cd-name</i> MESSAGE COUNT 	V	S		<i>cd-name</i>	N
ADD	<ul style="list-style-type: none"> Format 1 ADD <<i>id-1</i> <i>literal</i>> TO <<i>id-2</i> [ROUNDED]> [ON SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2 ADD <<i>id-1</i> <i>literal-1</i>> TO <<i>id-2</i> <i>literal-1</i>> GIVING <<i>id-3</i> [ROUNDED]> [ON SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3 ADD <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> [ROUNDED] [ON SIZE ERROR <i>imperative-stmt-1</i>] [NOT ON SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N
ALTER	<ul style="list-style-type: none"> Format ALTER <i>procedure-name-1</i> TO [PROCEED TO] <i>procedure-name-2</i> [<i>procedure-name-3</i> TO [PROCEED TO] <i>procedure-name-4</i>] 	V	S			N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
CALL	<ul style="list-style-type: none"> Format CALL <id-1 literal-1 procedure-pointer-1> [USING [< [BY REFERENCE] < id-2 literal-2 ADDRESS OF id-3 OMITTED > BY CONTENT < id-4 literal-3 LENGH OF id-5> BY VALUE < integer-1 [SIZE IS integer-2] LENGH OF id-7>>]] < GIVING RETURNING > < INTO id-8 ADDRESS OF id-9 > < [ON EXCEPTION imperative-stmt-1] [NOT ON EXCEPTION imperative-stmt-2] ON OVERFLOW imperative-stmt-1 > [END-CALL] 	V	S	id-1 id-2 id-3 id-4 id-5 id-7 id-8 id-9	id-2	N
CANCEL	<ul style="list-style-type: none"> Format CANCEL <id-1 literal-1> 	V	S			N
CHAIN	<ul style="list-style-type: none"> Format CHAIN <id-1 literal-1> CALL <id-1 literal-1 procedure-pointer-1> [USING [< [BY REFERENCE] < id-2 literal-2 ADDRESS OF id-3 OMITTED > BY CONTENT < id-4 literal-3 LENGH OF id-5> BY VALUE < integer-1 [SIZE IS integer-2] LENGH OF id-7>>]] [END-CHAIN] 	V	S	id-1 id-2 id-3 id-4	id-2	N
CLOSE	<ul style="list-style-type: none"> Format 1 CLOSE file-name-1 	V	S	file-name-1	file-name-1	N
	<ul style="list-style-type: none"> < < REEL UNIT > WITH LOCK < REEL UNIT > [< FOR REMOVAL WITH NO REWIND > WITH < NO REWIND LOCK >] 	VO	N	file-name-1	file-name-1	N
	<ul style="list-style-type: none"> Format 2 CLOSE filename-1 	V	S	file-name-1	file-name-1	N
	<ul style="list-style-type: none"> [<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND]] [WITH] <NO REWIND LOCK> 	VO	N	file-name-1	file-name-1	N
COMMIT	<ul style="list-style-type: none"> Format COMMIT 	V	N			N
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <id-1 [ROUNDED]> < EQUAL = > arithmetic-expression [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-COMPUTE] 	V	S	arithmetic-expr	id-1	S

1-170 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
CONTINUE	<ul style="list-style-type: none"> Format CONTINUE 	V	S			N
COPY	<ul style="list-style-type: none"> Format COPY <text-name external-file-name-literal > [< OF IN > <library-name library-name-literal>] [SUPPRESS] [REPLACING << ==pseudo-text-1== id-1 literal-1 word-1 > BY < ==pseudo-text-2== id-2 literal-2 word-2 >>] 	V	S			N/A
DELETE	<ul style="list-style-type: none"> Format DELETE file-name-1 RECORD [INVALID KEY imperative-stmt-1] [NOT INVALID KEY imperative-stmt-2] [END-DELETE] 	V	S	file-name-1	file-name-1	N
DELETE FILE	<ul style="list-style-type: none"> Format DELETE FILE < file-name > 	V	S	file-name		N
DISABLE	<ul style="list-style-type: none"> Format DISABLE < INPUT [TERMINAL] I-O TERMINAL OUTPUT > cd-name WITH KEY < id-1 literal-1 > 	V	S			N
DISPLAY	<ul style="list-style-type: none"> Format 1 DISPLAY <id-1 literal-1> [< id-2 literal-2 >] [UPON <mnemonic-name-1 function-name-1>] [WITH NO ADVANCING] < [ON EXCEPTION imperative-stmt-1] [NOT ON EXCEPTION imperative-stmt-2] [END-DISPLAY] > 	V	S	id-1		S
	<ul style="list-style-type: none"> Format 2 DISPLAY screen-name [< AT [LINE NUMBER < id-2 Integer-1 >] [< COLUMN COL > NUMBER < id-3 integer-2 >] AT < id-4 integer-3 > >] [END-DISPLAY] 	V	S	screen-name id-2 id-3		N
	<ul style="list-style-type: none"> Format 3 DISPLAY << id-1 literal-1 > > AT [LINE NUMBER < id-2 integer-1 >] [< COLUMN COL > NUMBER < id-3 integer-2 >] AT < id-4 integer-3 >] [UPON < CRT CRT-UNDER >] [MODE IS BLOCK] [FROM CRT] [MODE IS BLOCK] [WITH < < BELL BEEP > BLINK GRID HIGHLIGHT LOWLIGHT LEFTLINE OVERLINE REVERSE-VIDEO SIZE IS < id-5 integer-4 > UNDERLINE < FOREGROUND-COLOR FOREGROUND-COLOUR > IS integer-5 < BACKGROUND-COLOR BACKGROUND-COLOUR > IS integer-6 CONTROL IS < id-6 literal-2 > BLANK < SCREEN LINE > > >] [END-DISPLAY] 	V	S	id-1 id-2 id-3 id-5 id-6		N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
DIVIDE	<ul style="list-style-type: none"> Format 1 DIVIDE <id-1 literal-1> INTO <id-2 [ROUNDED]> [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2 DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING <id-3 [ROUNDED]> [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3 DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING <id-3 [ROUNDED]> [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4 DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING <id-3 [ROUNDED]> REMAINDER id-4 [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
	<ul style="list-style-type: none"> Format 5 DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING <id-3 [ROUNDED]> REMAINDER id-4 [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
ENABLE	<ul style="list-style-type: none"> Format ENABLE < INPUT [TERMINAL] I-O TERMINAL OUTPUT> cd-name WITH KEY < id-1 literal-1> 	V	S			N

1-172 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
END	<ul style="list-style-type: none"> Format END 	V	N			N
ENTER	<ul style="list-style-type: none"> Format ENTER <i>language-name</i> [<i>routine-name</i>] 	V	S			N
ENTRY	<ul style="list-style-type: none"> Format 1 ENTRY <i>literal-1</i> [USING < [BY REFERENCE] < <i>data-name-1</i> > BY VALUE < <i>data-name-2</i> >] 	V	S			N
	<ul style="list-style-type: none"> Format 2 ENTRY <i>literal-1</i> [<i>mnemonic-name</i>] [USING < [BY REFERENCE] < <i>data-name-1</i> [DELIMITED [BY SIZE]] <i>typedef-name-1</i> ANY > BY VALUE < <i>data-name-2</i> <i>typedef-name-2</i> ANY > > [REPEATED [<i>integer-1</i> TO <i>integer-2</i>]] [< GIVING RETURNING > < <i>data-name-3</i> <i>typedef-name-3</i> >] 	VO	N			N
EVALUATE	<ul style="list-style-type: none"> Format EVALUATE <<i>id-1</i> <i>literal-1</i> <i>expr-1</i> TRUE FALSE> [ALSO <<i>id-2</i> <i>literal-2</i> <i>expr-2</i> TRUE FALSE>] << WHEN < ANY <i>condition-1</i> TRUE FALSE [NOT] <<<i>id-3</i> <i>literal-5</i> <i>arithmetic-expression-3</i> > < THROUGH THRU > < <i>id-6</i> <i>literal-6</i> <i>arithmetic-expression-4</i>>] <i>partial-expression-2</i> >]> <i>imperative-statement-1</i>> [WHEN OTHER <i>imperative-statement-2</i>] [END-EVALUATE] 	V	S	<i>id-1</i> <i>expr-1</i> <i>id-2</i> <i>expr-2</i> <i>condition-1</i> <i>id-3</i> <i>id-4</i> <i>arithmetic-expr1</i> <i>arithmetic-expr2</i>		N
EXAMINE	<ul style="list-style-type: none"> Format 1 EXAMINE <i>id</i> REPLACING < ALL LEADING FIRST UNTIL FIRST> <i>literal-1</i> BY <i>literal-2</i> 	V	S	<i>id</i>	<i>id</i>	N
	<ul style="list-style-type: none"> Format 2 EXAMINE <i>id</i> TALLYNG < ALL LEADING UNTIL FIRST> <i>literal-1</i> [REPLACING BY] <i>literal-2</i> 	V	S	<i>id</i>	<i>id</i>	N
EXECUTE	<ul style="list-style-type: none"> Format EXEC[UTE] <i>text-name text-data</i> END-EXEC 	V	S	<i>text-name</i> <i>test-data</i>		N
EXHIBIT	<ul style="list-style-type: none"> Format 1 EXHIBIT < NAMED CHANGED NAMED CHANGED > < <i>id-1</i> <i>literal-1</i> > 	V	S	<i>id-1</i>		N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
EXIT	<ul style="list-style-type: none"> Format EXIT. 	V	S			N
EXIT METHOD	<ul style="list-style-type: none"> Format EXIT METHOD. 	V	N			N
EXIT PARAGRAPH	<ul style="list-style-type: none"> Format EXIT PARAGRAPH 	V	N			N
EXIT PERFORM	<ul style="list-style-type: none"> Format EXIT PERFORM [CYCLE] 	V	N			N
EXIT SECTION	<ul style="list-style-type: none"> Format EXIT SECTION 	V	N			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM [< GIVING RETURNING > < [ADDRESS OF] <i>id-1</i> <i>integer-1</i> >]. 	V	S	<i>id-1</i>		N
GENERATE	<ul style="list-style-type: none"> Format GENERATE < <i>data-name-1</i> <i>report-name-1</i> > 	VO	N			N
GOBACK	<ul style="list-style-type: none"> Format GOBACK [< GIVING RETURNING > < [ADDRESS OF] <i>id-1</i> <i>integer-1</i> >] 	V	S	<i>id-1</i>		N
GO TO	<ul style="list-style-type: none"> Format 1 GO TO [<i>procedure-name-1</i>] 	V	S			N
	<ul style="list-style-type: none"> Format 2 GO TO <i>procedure-name-1</i> [<i>procedure-name-2</i>] DEPENDING ON <i>id</i> 	V	S	<i>id-1</i>		N

1-174 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
IF	<ul style="list-style-type: none"> Format 1 IF <i>condition-1</i> THEN <<i>stmt-1</i> NEXT SENTENCE> [ELSE <<i>stmt-2</i> NEXT SENTENCE>] [END-IF] 	V	S	<i>condition-1</i>		N
	<ul style="list-style-type: none"> Format 2 IF <i>constant-name-1</i> [NOT] << > = > <i>literal-1</i> 	V	S			N
	<ul style="list-style-type: none"> Format 1 IF <i>constant-name-2</i> [NOT] DENIED 	N	N			N
INITIALIZE	<ul style="list-style-type: none"> Format INITIALIZE <i>id-1</i> [REPLACING <ALPHABETIC ALPHANUMERIC NUMERIC ALPHANUMERIC-EDITED NUMERIC-EDITED NATIONAL NATIONAL-EDITED DBCS > DATA BY <<i>id-2</i> <i>literal-1</i>>] 	V	S	<i>id-2</i>	<i>id-1</i>	N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING <<i>id-2</i> FOR < CHARACTERS [<BEFORE AFTER> INITIAL <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> INITIAL <<i>id-4</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> INITIAL <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> INITIAL <<i>id-4</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-5</i> <i>id-4</i> <i>id-3</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 4 INSPECT <i>id-1</i> CONVERTING <<i>id-6</i> <i>literal-4</i>> TO <<i>id-7</i> <i>literal-5</i>> [<BEFORE AFTER>] INITIAL <<i>id-4</i> <i>literal-2</i>> 	V	S	<i>d-1</i> <i>id-6</i> <i>id-7</i> <i>id-4</i>	<i>id-1</i>	N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
INVOKE	<ul style="list-style-type: none"> Format INVOKE <object-id-1 id-1 AS <template-1 OBJECT>> <literal-1 id-2> [USING <[BY REFERENCE] <[ADDRESS OF] id-4 id-3 literal-2> [BY CONTENT] < id-5 LENGTH OF id-6 literal-3 > [BY VALUE] <id-7 LENGTH OF id-6 integer-1 [SIZE IS integer-2]>>] [< RETURNING GIVING > <INTO id-9 ADDRESS OF id-10>] [[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [END-INVOKE] 	N	N			N
MERGE	<ul style="list-style-type: none"> Format MERGE file-name-1 <ON> <ASCENDING DESCENDING> KEY <data-name-1>> [COLLATING SEQUENCE IS alphabet-name-1] USING file-name-2 <file-name-3> <OUTPUT PROCEDURE IS procedure-name-1 [<THROUGH THRU> procedure-name-2] GIVING <file-name-4> > 	V	S	file-name-1 data-name-1 file-name-2 file-name-3	file-name-4	N
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <id-1 literal-1> TO <id-2> 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> id-1 TO <id-2> 	V	S	id-1	id-2	N
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <id-1 literal-1> BY <id-2 [ROUNDED]> [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2 MULTIPLY <id-1 literal-1> BY <id-2 literal-2> GIVING <id-3 [ROUNDED]> [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1 id-2	id-3	S
NEXT SENTENCE	<ul style="list-style-type: none"> Format NEXT SENTENCE 	V	S			N
NOTE	<ul style="list-style-type: none"> Format NOTE character-string 	V	S			N

1-176 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
ON	<ul style="list-style-type: none"> Format ON < <i>literal-1</i> <i>id-1</i> > [AND EVERY < <i>literal-2</i> <i>id-2</i> >] [UNTIL < <i>literal-3</i> <i>id-3</i> >] < <i>imperative-statement-1</i> NEXT SENTENCE > [< ELSE OTHERWISE > < <i>imperative-statement-2</i> NEXT SENTENCE >] 	V	S	id-1 id-2 id-3		N
OPEN	<ul style="list-style-type: none"> Format OPEN << INPUT < <i>file-name-1</i> [WITH LOCK]> OUTPUT < <i>file-name-2</i> [WITH LOCK] I-O < <i>file-name-3</i> [WITH LOCK]> EXTEND < <i>file-name-4</i> [WITH LOCK]>> 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
	OPEN << INPUT < <i>file-name-1</i> > OUTPUT < <i>file-name-2</i> I-O < <i>file-name-3</i> > EXTEND < <i>file-name-4</i> >>	VO	N			N
PERFORM	<ul style="list-style-type: none"> Format 1 PERFORM [<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>]] [<i>imperative-stmt-1</i> [END-PERFORM]] 	V	S			N
	<ul style="list-style-type: none"> Format 2 PERFORM [<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>]] <<i>id-1</i> <i>integer-1</i>> TIMES [<i>imperative-stmt-1</i> END-PERFORM] 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3 PERFORM [<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>]] [WITH TEST <BEFORE AFTER>] UNTIL < <i>condition-1</i> EXIT > [<i>imperative-stmt-1</i> [END-PERFORM]] 	V	S	condition-1		N
	<ul style="list-style-type: none"> Format 4 PERFORM [<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>]] [WITH TEST <BEFORE AFTER>] VARYING <<i>id-2</i> <i>index-name-1</i>> FROM <<i>id-3</i> <i>index-name-2</i> <i>literal-1</i>> BY <<i>id-4</i> <i>literal-2</i>> UNTIL <<i>condition-1</i>> [AFTER <<i>id-5</i> <i>index-name-3</i>> FROM <<i>id-6</i> <i>index-name-4</i> <i>literal-3</i>> BY <<i>id-7</i> <i>literal-4</i>> UNTIL <i>condition-2</i>] [<i>imperative-stmt-1</i> [END-PERFORM]] 	V	S	id-2 index-name-1 id-3 index-name-2 id-4 condition-1 id-5 index-name-3 id-6 index-name-4 id-7 condition-2	id-2 index-name-1	N
PURGE	<ul style="list-style-type: none"> Format PURGE <i>cd-name</i> 	V	S	od-name		N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
READ	<ul style="list-style-type: none"> Format 1 READ <i>file-name</i> [NEXT] RECORD [INTO <i>id</i>] [WITH < [< KEPT NO >] LOCK WAIT>] [AT END <i>imperative-stmt-1</i>] [NOT AT END <i>imperative-stmt-2</i>] [END-READ] 	V	S	file-name	id	N
	<ul style="list-style-type: none"> Format 2 READ <i>file-name</i> [NEXT] RECORD [INTO <i>id</i>] [AT END <i>imperative-stmt-1</i>] [NOT AT END <i>imperative-stmt-2</i>] [END-READ] 	V	S	file-name	id	N
	<ul style="list-style-type: none"> Format 3 READ <i>file-name</i> [< NEXT PREVIOUS >] RECORD [INTO <i>id</i>] [WITH < [< KEPT NO IGNORE >] LOCK WAIT>] [AT END <i>imperative-stmt-1</i>] [NOT AT END <i>imperative-stmt-2</i>] [END-READ] 	V	S	file-name	id	N
	<ul style="list-style-type: none"> Format 4 READ <i>file-name</i> RECORD [INTO <i>id-1</i>] [WITH < [< KEPT NO >] LOCK WAIT>] [INVALID KEY <i>imperative-stmt-3</i>] [NOT INVALID KEY <i>imperative-stmt-4</i>] [END-READ] 	V	S	file-name	id	N
	<ul style="list-style-type: none"> Format 5 READ <i>file-name</i> RECORD [INTO <i>id-1</i>] [WITH < [< KEPT NO IGNORE >] LOCK WAIT>] [KEY IS < <i>data-name</i> <i>split-key-name</i> >] [INVALID KEY <i>imperative-stmt-3</i>] [NOT INVALID KEY <i>imperative-stmt-4</i>] [END-READ] 	V	S	file-name data-name split-key-name	id	N
READY TRACE	<ul style="list-style-type: none"> Format READY TRACE 	VO	N			N
RECEIVE	<ul style="list-style-type: none"> Format RECEIVE <i>cd-name</i> < MESSAGE SEGMENT > 	V	S	cd-name		N
	<ul style="list-style-type: none"> INTO <i>id-1</i> [NO DATA <i>imperative stmt-1</i>] [WITH DATA <i>imperative stmt-2</i>] [END-RECEIVE] 	VO	N			N

1-178 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
RELEASE	<ul style="list-style-type: none"> Format RELEASE <i>record-name</i> [FROM <i>id</i>] 	V	S	<i>record-name</i> <i>id</i>	<i>record-name</i>	N
REPLACE	<ul style="list-style-type: none"> Format 1 REPLACE <==<i>pseudo-text</i>==BY==<i>pseudo-text-2</i>==> 	V	S			N/ A
	<ul style="list-style-type: none"> Format 2 REPLACE OFF 	V	S			N/ A
RESET TRACE	<ul style="list-style-type: none"> Format RESET TRACE 	VO	N			N
RETURN	<ul style="list-style-type: none"> Format RETURN <i>file-name</i> RECORD [INTO <i>id-1</i>] AT END <i>imperative-stmt-1</i> [NOT AT END <i>imperative-stmt-2</i>] [END-RETURN] 	V	S	<i>file-name</i>	<i>id-1</i>	N
REWRITE	<ul style="list-style-type: none"> Format 1 REWRITE <i>record-name</i> [FROM <i>id</i>] [END-REWRITE] 	V	S	<i>record-name</i> <i>id-1</i>	<i>record-name</i>	N
	<ul style="list-style-type: none"> Format 2 REWRITE <i>record-name</i> [FROM <i>id</i>] [INVALID KEY <i>imperative-stmt-1</i>] [NOT INVALID KEY <i>imperative-stmt-2</i>] [END-REWRITE] 	V	S	<i>file-name-1</i> <i>id-1</i>	<i>file-name-1</i>	N
ROLLBACK	<ul style="list-style-type: none"> Format ROLLBACK 	VO	N			N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
SEARCH	<ul style="list-style-type: none"> Format 1 SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [AT END <i>imperative-stmt-1</i>] <WHEN <i>condition-1</i> [<<i>imperative-stmt-2</i> NEXT SENTENCE>] [END-SEARCH] 	V	S	<i>id-1</i> <i>condition-1</i>	<i>id-2</i> <i>index-name-1</i>	N
	<ul style="list-style-type: none"> Format 2 SEARCH ALL <i>id-1</i> [AT END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> IS < EQUAL TO = > <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> IS < EQUAL TO = > <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] [<<i>imperative-stmt-2</i> NEXT SENTENCE>] [END-SEARCH] 	V	S	<i>id-1</i> <i>data-name-1</i> <i>id-3</i> <i>data-name-2</i> <i>id-4</i> <i>condition-name-1</i> <i>condition-name-2</i>		N
SEND	<ul style="list-style-type: none"> Format 1 SEND <i>cd-name</i> FROM <i>id-1</i> 	VO	N			N
	<ul style="list-style-type: none"> Format 2 SEND <i>cd-name</i> [FROM <i>id-1</i>] < WITH < <i>id-2</i> ESI EMI EGI > [< BEFORE AFTER > ADVANCING < <<i>id-1</i> <i>integer</i>> [LINE LINES] < <i>mnemonic name</i> PAGE > >] 	VO	N			N
SERVICE	<ul style="list-style-type: none"> Format SERVICE < LABEL RELOAD <i>id</i> > 	VO	N			N
SET	<ul style="list-style-type: none"> Format 1 SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-6</i> <i>integer-2</i>> 	V	S	<i>index-name-2</i> <i>id-6</i>	<i>index-name-1</i> <i>id-1</i>	N
	<ul style="list-style-type: none"> Format 2 SET <<i>pointer-name-3</i>> <UP DOWN > BY <<i>id-3</i> <i>integer-1</i> LENGTH OF <i>id-4</i>> 	V	S	<i>pointer-name-3</i> <i>id-3</i> <i>id-4</i>	<i>pointer-name-3</i>	N
	<ul style="list-style-type: none"> Format 3 SET <<<i>mnemonic-name-1</i>>TO <ON OFF>> 	V	S		<i>mnemonic-name-1</i>	N

1-180 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
SET	<ul style="list-style-type: none"> Format 4 SET <<condition-name-1> TO < TRUE FALSE >> 	V	S		condition-variable	S
	<ul style="list-style-type: none"> Format 5 SET <pointer-name-1 ADDRESS OF id-5> TO <pointer-name-2 ADDRESS OF id-7 NULL NULLS> 	V	S	pointer-name-2 id-7	pointer-name-1 id-5	N
	<ul style="list-style-type: none"> Format 6 SET <procedure-pointer-name-1> TO <procedure-pointer-name-2 ENTRY <id-8 literal-1> NULL NULLS > 	V	N			N
	<ul style="list-style-type: none"> Format 7 SET <index-name-3> < UP DOWN > BY <id-7 integer-3> 	V	S	index-name-3 id-7	index-name-3	N
	<ul style="list-style-type: none"> Format 8 SET <object-id-1>TO <object-id-2 NULL> 	V	N			N
SORT	<ul style="list-style-type: none"> Format 1 SORT file-name-1 ON <<ASCENDING DESCENDING> KEY <data-name-1>> [WITH DUPLICATES IN ORDER] <USING <file-name-2> INPUT PROCEDURE IS procedure-name-1 [<THROUGH THRU> procedure-name-2]> <GIVING <file-name-3> OUTPUT PROCEDURE IS procedure-name-3 [<THROUGH THRU> procedure-name-4]> 	V	S	file-name-1 data-name-1 file-name-2	file-name-3	N
	<ul style="list-style-type: none"> Format 1 SORT file-name-1 ON <<ASCENDING DESCENDING> KEY <data-name-1>> [WITH DUPLICATES IN ORDER] [COLLATING SEQUENCE IS alphabet-name] <USING <file-name-2> INPUT PROCEDURE IS procedure-name-1 [<THROUGH THRU> procedure-name-2]> <GIVING <file-name-3> OUTPUT PROCEDURE IS procedure-name-3 [<THROUGH THRU> procedure-name-4]> 	VO	N			N
	<ul style="list-style-type: none"> Format 2 SORT data-name-2 ON <<ASCENDING DESCENDING> KEY <data-name-1>> [WITH DUPLICATES IN ORDER] 	V	S	data-name-2 data-name-1	data-name-2	N
	<ul style="list-style-type: none"> Format 1 SORT data-name-2 ON <<ASCENDING DESCENDING> KEY <data-name-1>> [WITH DUPLICATES IN ORDER] [COLLATING SEQUENCE IS alphabet-name] 	VO	N			N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
START	<ul style="list-style-type: none"> Format 1 START <i>file-name-1</i> [KEY IS <EQUAL TO = LESS THAN < GREATER THAN > NOT LESS THAN NOT < NOT GREATER THAN NOT > LESS THAN OR EQUAL TO <= GREATER THAN OR EQUAL TO >= > < <i>data-name-1</i> <i>split-key-name-1</i> > [WITH SIZE < <i>id-1</i> <i>literal-1</i>>]] [INVALID KEY <i>imperative-stmt-1</i>] [NOT INVALID KEY <i>imperative-stmt-2</i>] [END-START] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	<ul style="list-style-type: none"> Format 2 START <i>file-name-1</i> [KEY IS <EQUAL TO = LESS THAN < GREATER THAN > NOT LESS THAN NOT < NOT GREATER THAN NOT > LESS THAN OR EQUAL TO <= GREATER THAN OR EQUAL TO >= > <i>data-name-1</i>] [INVALID KEY <i>imperative-stmt-1</i>] [NOT INVALID KEY <i>imperative-stmt-2</i>] [END-START] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format 1 STOP <RUN <i>literal-1</i>> 	V	S			N
	<ul style="list-style-type: none"> Format 2 STOP RUN [< GIVING RETURNING > < [ADDRESS OF] <i>id-1</i> integer-1 [SIZE IS <i>integer-2</i>]>] 	VO	N			N
STRING	<ul style="list-style-type: none"> Format STRING <<<i>id-1</i> <i>literal-1</i>> DELIMITED [BY] <<i>id-2</i> <i>literal-2</i> SIZE>> INTO <i>id-3</i> [WITH POINTER <i>id-4</i>] [ON OVERFLOW <i>imperative-stmt-1</i>] [NOT ON OVERFLOW <i>imperative-stmt-2</i>] [END-STRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>	<i>id-3</i> <i>id-4</i>	S

1-182 Supported COBOL Statements
Supported MicroFocus COBOL statements

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <id-1 literal-1> FROM <id-2 [ROUNDED]> [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2 SUBTRACT <id-1 literal-1> FROM <id-2 literal-2> GIVING <id-3 [ROUNDED]> [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> id-1 FROM id-2 [ROUNDED] [ON SIZE ERROR imperative-stmt-1] [NOT ON SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	N
SUPPRESS	<ul style="list-style-type: none"> Format SUPPRESS PRINTING 	VO	N			N
TERMINATE	<ul style="list-style-type: none"> Format TERMINATE <report-name-1> 	VO	N			N
TRANSFORM	<ul style="list-style-type: none"> Format TRANSFORM id-3 CHARACTERS FROM < figurative-constant-1 nonnumeric literal-1 id-1 > TO < figurative-constant-2 nonnumeric-literal-2 id-2 > 	VO	N			N
UNLOCK	<ul style="list-style-type: none"> Format UNLOCK file-name <RECORD RECORDS> 	VO	N			N

Statements	Format	Parser	CE			
			Gen.	Read Variables	Written Variables	DB
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING <i>id-1</i> [DELIMITED BY [ALL] <<i>id-2</i> <i>literal-1</i>> [OR [ALL] <<i>id-3</i> <i>literal-2</i>>]] INTO <<i>id-4</i> [DELIMITER IN <i>id-5</i>] [COUNT IN <i>id-6</i>]> [WITH POINTER <i>id-7</i>] [TALLYING IN <i>id-8</i>] [ON OVERFLOW <i>imperative-stmt-1</i>] [NOT ON OVERFLOW <i>imperative-stmt-2</i>] [END-UNSTRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-7</i> <i>id-8</i>	<i>id-4</i> <i>id-5</i> <i>id-6</i> <i>id-7</i> <i>id-8</i>	S
USE	<ul style="list-style-type: none"> Format 1 USE [GLOBAL] AFTER STANDARD < EXCEPTION ERROR > PROCEDURE ON <[<i>file-name-1</i>] INPUT OUTPUT I-O EXTEND > 	VO	N			N
	<ul style="list-style-type: none"> Format 2 USE [GLOBAL] BEFORE REPORTING <i>id-1</i> 	VO	N			N
	<ul style="list-style-type: none"> Format 3 USE FOR DEBUGGING ON < <i>cd-name-1</i> [ALL REFERENCES OF] <i>id-1</i> <i>file-name-1</i> <i>procedure-name-1</i> ALL PROCEDURES > 	VO	N			N
WRITE	<ul style="list-style-type: none"> Format 1 WRITE <i>record-name</i> [< FROM <i>id-1</i> FROM <i>literal</i> >] [INVALID KEY <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>record-name</i> <i>id-1</i>	<i>record-name</i>	N
	<ul style="list-style-type: none"> Format 2 WRITE <i>record-name</i> [< FROM <i>id-1</i> FROM <i>literal</i> >] [< BEFORE AFTER > ADVANCING < < <i>id-2</i> <i>integer</i> > [LINE LINES] <i>mnemonic-name</i> PAGE TAB FRMFEED >] [AT <END-OF-PAGE EOP> <i>imperative-stmt-1</i>] [NOT AT <END-OF-PAGE EOP> <i>imperative-stmt-2</i>] [END-WRITE] 	N	N			N

1-184 Supported COBOL Statements
Supported MicroFocus COBOL statements

Intrinsic functions

Functions	Parser	CE	
		Gen.	DB
ABS	V	N	N
ACOS	V	N	N
ANNUITY	V	N	N
ASIN	V	N	N
ATAN	V	N	N
CHAR	V	N	N
CHAR-NATIONAL	V	N	N
COS	V	N	N
CURRENT-DATE	V	N	N
DATE-OF-INTEGER	V	N	N
DAY-OF-INTEGER	V	N	N
DAY-TO-YYYYDDD	V	N	N
DISPLAY-OF	V	N	N
E	V	N	N
EXP	V	N	N
EXP10	V	N	N
FACTORIAL	V	N	N
FRACTION-PART	V	N	N
INTEGER	V	N	N
INTEGER-OF-DATE	V	N	N
INTEGER-OF-DAY	V	N	N

Functions	Parser	CE	
		Gen.	DB
INTEGER-PART	V	N	N
LENGTH	V	N	N
LENGTH-AN	V	N	N
LOG	V	N	N
LOG10	V	N	N
LOWER-CASE	V	N	N
MAX	V	N	N
MEAN	V	N	N
MEDIAN	V	N	N
MIDRANGE	V	N	N
MIN	V	N	N
MOD	V	N	N
NATIONAL-OF	V	N	N
NUMVAL	V	N	N
NUMVAL-C	V	N	N
ORD	V	N	N
ORD-MAX	V	N	N
ORD-MIN	V	N	N
P1	V	N	N
PRESENT-VALUE	V	N	N
RANDOM	V	N	N
RANGE	V	N	N

1-186 Supported COBOL Statements
Supported MicroFocus COBOL statements

Functions	Parser	CE	
		Gen.	DB
REM	V	N	N
REVERSE	V	N	N
SIGN	V	N	N
SIN	V	N	N
SQRT	V	N	N
STANDARD-DEVIATION	V	N	N
SUM	V	N	N
TAN	V	N	N
UPPER-CASE	V	N	N
VARIANCE	V	N	N
WHEN-COMPILED	V	N	N
YEARWINDOW	V	N	N

Supported UTS 4000 COBOL statements

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S

Reserved data names

Data names	Parser	CE
STATION-ID	V	
ATTENTION-INDEX	V	
BACK-TAB	V	
FUNCTION-INDEX	V	
CURRENT-FIELD	V	
FIELD-CHANGED-FLAG	V	
LINE-INDEX	V	
COLUMN-INDEX	V	

1-188 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Data names	Parser	CE
FIELD-EXIT-KEY	V	

Literals

Literal type	Format	Parser	CE
Nonnumeric literal	<ul style="list-style-type: none"> Format Example: "THIS ISN""T WRONG" 	V	S
Numeric literals	<ul style="list-style-type: none"> Format Example: -2.71828 	V	S
Hexadecimal literals	<ul style="list-style-type: none"> Format Example: H"FC1D" 	V	S
Octal literals	<ul style="list-style-type: none"> Format Example: O"7063" 	V	S
Binary literals	<ul style="list-style-type: none"> Format Example: B"0100101101010" 	V	S
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

COBOL program structure

Program type	Format	Parser	CE
Single program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> END-PROGRAM <i>program-name-1</i> .	V	S

Identification Division

Paragraphs	Format	Parser	CE
Program Identification Division	<ul style="list-style-type: none"> Format IDENTIFICATION DIVISION. PROGRAM-ID. <i>program-name</i> . [AUTHOR. <i>[comment-entry]</i>] [INSTALLATION. <i>[comment-entry]</i>] [DATE-WRITTEN. <i>[comment-entry]</i>] [SECURITY. <i>[comment-entry]</i>]	V	S

1-190 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Environment Division

Input-output section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential organization FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN <i>assignment-name-1</i>	V	S
	[; ORGANIZATION [IS] SEQUENTIAL] [; NO DELETE [OF <i>literal</i>]] [; [FILE] STATUS [IS] <i>data-name-1</i>] [; RESERVE [NO] [ALTERNATE] AREA] [; ALTERNATE [INPUT AREA] STATUS [IS] <i>data-name-3</i>]	VO	N
	<ul style="list-style-type: none"> Format 2: relative organization FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN <i>assignment-name-1</i>	V	S
	[; ORGANIZATION [IS] SEQUENTIAL] [; NO DELETE [OF <i>literal</i>]] [; [FILE] STATUS [IS] <i>data-name-1</i>] [; RESERVE [NO] [ALTERNATE] AREA] [; ALTERNATE [INPUT AREA] STATUS [IS] <i>data-name-3</i>]	VO	N
I-O-CONTROL paragraph	<ul style="list-style-type: none"> Format I-O-CONTROL. [; SAME [RECORD] [AREA] [FOR] <i>file-name-1</i> [, <i>file-name-4</i>]]	VO	N

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COMPUTER paragraph	<ul style="list-style-type: none"> Format OBJECT-COMPUTER. <UTS40 UTS4000 UTS30> [, SEGMENT LIMIT IS <i>integer-1</i>] [MEMORY SIZE <i>integer</i> <CHARACTRERS MODULES WORDS>] [PROGRAM COLLATING SEQUENCE IS <i>alphabet-name</i>] [SUSPEND-KEY IS DISABLED] 	VO	
SOURCE-COMPUTER paragraph	<ul style="list-style-type: none"> Format SOURCE-COMPUTER. [<i>computer-name</i> [WITH DEBUGGING MODE].] 	VO	N
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> Format SPECIAL-NAMES. [<i>alphabet-name-1</i> [IS] <STANDARD-1 NATIVE ASCII>] 	V	S
	[, CURRENCY [SIGN] [IS] <i>literal-6</i> [[WITH] PICTURE SYMBOL <i>literal-7</i>]]	VO	N
	[, DECIMAL-POINT [IS] COMMA] [.]	V	S

Data Division

Entries and clauses	Format	Parser	CE
Data Division	<ul style="list-style-type: none"> Format DATA DIVISION. [FILE SECTION. [<i>file-description-entry</i> <i>record-description-entry</i>]] [SCREEN SECTION. [<i>screen-description-entry</i>]] [WORKING-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [SHARED-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [LINKAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [COMMON-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] 	V	S

1-192 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Entries and clauses	Format	Parser	CE
File Description (FD) Entry	<ul style="list-style-type: none"> • Format FD <i>file-name-1</i> [; BLOCK [CONTAINS] <i>integer-1</i> <CHARACTERS RECORDS>] [; RECORD [CONTAINS] <i>integer-2</i> TO <i>integer-3</i> [CHARACTERS]] [; LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED> VALUE OF <FILE-ID PERIPHERAL-ID> [IS] <<i>data-name-1</i> <i>literal-1</i>> [; DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-2</i> [, <i>data-name-3</i> ...]] [; CODE-SET [IS] <i>alphabet-name</i>]. 	V	S
Screen Description (SR) Entry	<ul style="list-style-type: none"> • Format SR <i>file-name</i> [; DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-1</i> [, <i>data-name-2</i> ...]] [; RECORD [CONTAINS] <i>integer-1</i> TO <i>integer-2</i> [CHARACTERS]] [; LINAGE IS <i>integer-3</i> LINES] [; LINE SIZE IS <i>integer-4</i> CHARACTERS] 	V	S
SCREEN DATA description	<ul style="list-style-type: none"> • Format 1 <i>level-number</i> <<i>data-name-1</i> FILLER> [occurs-clause] [picture-clause] [usage-clause] [justified-clause] [value-clause] [sign-clause] [protected-clause] [intensity-clause] [emphasis-clause] [line-clause] [column-clause] [alternate-character-clause] [upper-case-clause] 	V	S
	<ul style="list-style-type: none"> • Format 2 88 <i>condition-name</i> <i>value-clause</i>. 	V	S
Data Description Entry	<ul style="list-style-type: none"> • Format 1 <i>level-number</i> <<i>data-name-1</i> FILLER> [redefines-clause] [blank-when-zero-clause] [justified-clause] [occurs-clause] [picture-clause] [sign-clause] [synchronized-clause] [usage-clause] [value-clause] 	V	S
	<ul style="list-style-type: none"> • Format 2 88 <i>condition-name</i> <i>value-clause</i>. 	V	S
ALTERNATE CHARACTER clause	<ul style="list-style-type: none"> • Format ALTERNATE [CHARACTER] REPRESENTATION 	V	S
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> • Format ; BLANK [WHEN] ZERO 	V	S

Entries and clauses	Format	Parser	CE
COLUMN clause	<ul style="list-style-type: none"> Format <COL COLUMN> NUMBER [IS] <integer-1 PLUS integer-2> 	VO	N
EMPHASIS clause	<ul style="list-style-type: none"> Format EMPHASIS [IS] <UNDERSCORE STRIKE-THROUGH FRAMED COLUMN-SEPARATOR> 	VO	N
INTENSITY clause	<ul style="list-style-type: none"> Format INTENSITY [IS] <BLINK HIGH LOW NON-DISPLAYED REVERSE> 	VO	N
JUSTIFIED clause	<ul style="list-style-type: none"> Format <JUSTIFIED JUST> [RIGHT] 	VO	N
LINE clause	<ul style="list-style-type: none"> Format LINE [NUMBER] [IS] <integer-1 PLUS integer-2> 	VO	N
OCCURS clause	<ul style="list-style-type: none"> Format OCCURS integer-2 [TIMES] [INDEXED [BY] index-name-1] 	V	S
PICTURE clause	<ul style="list-style-type: none"> Format <PICTURE PIC> [IS] character-string 	V	S
PROTECTED clause	<ul style="list-style-type: none"> Format <PROT PROTECTED> 	VO	N
REDEFINES clause	<ul style="list-style-type: none"> Format level-number data-name-1 REDEFINES data-name-2 	V	S
SIGN clause	<ul style="list-style-type: none"> Format SIGN [IS] <LEADING TRAILING> [SEPARATE CHARACTER] 	V	S
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZED SYNC> [LEFT RIGHT] 	VO	N
UPPER-CASE clause	<ul style="list-style-type: none"> Format UPPER-CASE 	VO	N

1-194 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Entries and clauses	Format	Parser	CE
USAGE clause	<ul style="list-style-type: none"> Format USAGE [IS] <COMP COMP-1 COMP-2 COMPUTATIONAL DISPLAY INDEX LOCK> 	V	S
VALUE clause	<ul style="list-style-type: none"> Format 1: literal value VALUE [IS] <i>literal</i> 	V	S
	<ul style="list-style-type: none"> Format 2: condition-name value 88 <i>condition-name-1</i> <VALUE [IS] VALUES [ARE]> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]. 	V	S

Procedure Division

Arithmetic operators

Binary operation	Meaning	Format	Parser	CE			
				Gen.	Read variables	Written variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S

+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

Conditional Expressions

Conditions	Format	Parser	CE			
			Gen.	Read variables	Written variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1 [IS] [NOT] <NUMERIC ALPHABETIC></i> 	V	S	<i>id-1</i>		S
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format <i>operand-1 [IS] [NOT] <GREATER [THAN] LESS [THAN] EQUAL > < => operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>		S
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1 <AND OR> condition-2</i> 	V	S	<i>condition-1</i> <i>condition-2</i>		S

Statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format PROCEDURE DIVISION [USING <i>data-name-1</i>]. 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N

1-196 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DECLARATIVES section	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. <i>paragraph-name</i>. [<i>sentences</i>] END DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: elementary ACCEPT <i>id</i> [NON-WAIT] 	V				N
	<ul style="list-style-type: none"> Format 2: group ACCEPT <<i>id</i> SCREEN> AT <i>id-2</i> 	V				N
	<ul style="list-style-type: none"> Format 3: sequenced ACCEPT <i>id-1</i>, <i>id-2</i> [, <i>id-3</i> ...] 	V				N
	<ul style="list-style-type: none"> Format 4 ACCEPT <i>id</i> FROM <DATE DAY TIME> 	V				N
	<ul style="list-style-type: none"> Format 5 ACCEPT DATE FROM <i>id-1</i> 	V				N
ADD	<ul style="list-style-type: none"> Format 1 ADD <<i>id-1</i> <i>literal</i>> TO <i>id-2</i> [ROUNDED] [: [ON] SIZE ERROR <i>imperative-stmt-1</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <<i>id-1</i> <i>literal-1</i>> TO <<i>id-2</i> <i>literal-1</i>> GIVING <i>id-3</i> [ROUNDED] [: [ON] SIZE ERROR <i>imperative-stmt-1</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
ALERT	<ul style="list-style-type: none"> Format ALERT [ON OFF] 	V				N
ALTER	<ul style="list-style-type: none"> Format ALTER <i>procedure-name-1</i> TO [PROCEED TO] <i>procedure-name-2</i> 	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
CALL	<ul style="list-style-type: none"> Format 1 CALL <i>literal-1</i> [USING <i>data-name-1</i>] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i>	<i>id-2</i>	S
CHANGE	<ul style="list-style-type: none"> Format 1 CHANGE <i>id-1</i> TO <HIGH LOW BLINK NON-DISPLAYED> [INTENSITY [WITH [NO] TABSET]] 	V				N
	<ul style="list-style-type: none"> Format 2 CHANGE <i>id-1</i> TO <NO UNDERSCOPE STRIKE-THROUGH COLUMN-SEPARATOR FRAMED ALTERNATE> EMPHASIS 	V				N
	<ul style="list-style-type: none"> Format 3 CHANGE <i>id-1</i> TO [NO] SUBSET 	V				N
CLEAR	<ul style="list-style-type: none"> Format 1 CLEAR <<i>file-name</i> <i>record-name</i>> 	V				N
	<ul style="list-style-type: none"> Format 2 CLEAR SCREEN 					N
	<ul style="list-style-type: none"> Format 3 CLEAR ACCEPT 	V				N
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> [<REEL UNIT> [WITH NO REWIND]] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: relative CLOSE <i>file-name-1</i> 	V	S			<i>file-name-1</i>
			VO	N		
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <i>id-1</i> [ROUNDED] = <i>arithmetic-expr</i> [: [ON] SIZE ERROR <i>imperative-stmt-1</i>] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S

1-198 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
COPY	<ul style="list-style-type: none"> Format COPY <i>text-name</i>. 	V				
DELETE	<ul style="list-style-type: none"> Format DELETE <i>file-name-1</i> [RECORD] [INVALID [KEY] <i>imperative-stmt-1</i>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISPLAY	<ul style="list-style-type: none"> Format DISPLAY <<i>id-1</i> <i>literal-1</i>> [AT LINE <i>literal-2</i> [, COLUMN <i>literal-4</i>]] 	V				
DIVIDE	<ul style="list-style-type: none"> Format 1 DIVIDE <<i>id-1</i> <i>literal-1</i>> INTO <i>id-2</i> [ROUNDED] [; [ON] SIZE ERROR <i>imperative-stmt-1</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING DIVIDE <<i>id-1</i> <i>literal-1</i>> <BY INTO> <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [REMAINDER <i>id-4</i>] [; [ON] SIZE ERROR <i>imperative-stmt-1</i>] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
EXIT	<ul style="list-style-type: none"> Format 1 EXIT. 	V	S			N
	<ul style="list-style-type: none"> Format 2 EXIT <i>procedure-name</i>. 	V				N
EXIT PRO-GRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
FORMAT	<ul style="list-style-type: none"> Format FORMAT <i>record-name</i> 	V	S			N
FUNCTION NUMVAL	<ul style="list-style-type: none"> Format FUNCTION NUMVAL (<i>alphanumeric-argument</i>) 	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO [TO] <i>procedure-name-1</i> 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO [TO] <i>procedure-name-1</i> DEPENDING [ON] <i>id-1</i> 	V	S	<i>id-1</i>		N
IF	<ul style="list-style-type: none"> Format IF <i>condition-1</i> THEN <<i>stmt-1</i> NEXT SENTENCE> [; ELSE <<i>stmt-2</i> NEXT SENTENCE>] 	V	S	<i>condition-1</i>		N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING, <i>id-2</i> FOR, <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] , <ALL LEADING FIRST>, <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-5</i> <i>id-4</i> <i>id-3</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING, <i>id-2</i> FOR, <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] , <ALL LEADING FIRST>, <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
LOAD	<ul style="list-style-type: none"> Format LOAD <<i>id-1</i> <i>literal-1</i>> FROM <<i>id-2</i> <i>literal-2</i>> 	V	S			N
LOCK	<ul style="list-style-type: none"> Format LOCK <i>id-1</i> [<i>id-2</i>] 	V	S			N
MOVE	<ul style="list-style-type: none"> Format MOVE <<i>id-1</i> <i>literal-1</i>> <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	S

1-200 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <id-1 literal-1> BY id-2 [ROUNDED] [: [ON] SIZE ERROR imperative-stmt-1] 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [: [ON] SIZE ERROR imperative-stmt-1] 	V	S	id-1 id-2	id-3	S
OPEN	<ul style="list-style-type: none"> Format OPEN <INPUT file-name-1 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
	[[WITH] NO REWIND]	VO	N			
	OUTPUT file-name-2 [V	S			
	[[WITH] NO REWIND]	VO	N			
	I-O file-name-3 EXTEND file-name-4>	V	S			
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] <id-1 integer-1> TIMES 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase and END-PERFORM PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] UNTIL condition-1 	V	S	condition-1		N
READ	<ul style="list-style-type: none"> Format 1 READ file-name-1 RECORD [INTO id] [: [AT] END imperative-stmt] 	V	S			N
	<ul style="list-style-type: none"> Format 2 READ file-name-1 RECORD [INTO id] ; SEARCH KEY IS data-name [: INVALID [KEY] imperative-stmt] 	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
	<ul style="list-style-type: none"> Format 3 READ <i>file-name-1</i> NEXT RECORD [INTO <i>id</i>] [: [AT] END <i>imperative-stmf</i>] 	V	S			N
	<ul style="list-style-type: none"> Format 4 READ <i>file-name-1</i> RECORD [INTO <i>id</i>] [: INVALID [KEY] <i>imperative-stmf</i>] 	V	S	<i>file-name-1</i> <i>data-name-1</i>	<i>file-name-1</i> <i>id-1</i>	N
REWRITE	<ul style="list-style-type: none"> Format REWRITE <i>file-name-1</i> [FROM <i>id-1</i>] [: INVALID [KEY] <i>imperative-stmf</i>] 	V	S	<i>file-name-1</i> <i>id-1</i>	<i>file-name-1</i>	N
SEEK	<ul style="list-style-type: none"> Format SEEK <i>file-name</i> [RECORD] KEY [IS] <i>data-name</i> [: INVALID [KEY] <i>imperative-stmf</i>] 	V	S			N
SET	<ul style="list-style-type: none"> Format 1 SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-2</i> <i>integer-1</i>> 	V	S	<i>index-name-2</i> <i>id-2</i>	<i>index-name-1</i> <i>id-1</i>	N
	<ul style="list-style-type: none"> Format 2 SET <i>index-name-3</i> <UP BY DOWN BY> <<i>id-3</i> <i>integer-2</i>> 	V	S	<i>index-name-3</i> <i>id-3</i>	<i>index-name-3</i>	N
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> [[WITH] DUPLICATES [IN] [ORDER]] 	V	S			N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N			N
	<USING <i>file-name-2</i> INPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i> > <GIVING <i>file-name-3</i> OUTPUT PROCEDURE [IS] <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i> >	V	S			N
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [KEY [IS] <EQUAL [TO] = GREATER [THAN] > NOT LESS [THAN] NOT <> <i>data-name</i>] [: INVALID [KEY] <i>imperative-stmt-1</i>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N

1-202 Supported COBOL Statements
Supported UTS 4000 COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <id-1 literal-1> FROM id-2 [ROUNDED] [: [ON] SIZE ERROR imperative-stmt-1] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <id-1 literal-1> FROM <id-2 literal-2> GIVING id-3 [ROUNDED] [: [ON] SIZE ERROR imperative-stmt-1] 	V	S	id-1 id-2	id-3	S
UNLOCK	<ul style="list-style-type: none"> Format UNLOCK id-1 [, id-2 ...] 	V	S			N
USE	<ul style="list-style-type: none"> Format 1 USE AFTER [STANDARD] <ERROR EXCEPTION> PROCEDURE [ON] <file-name INPUT OUTPUT I-O> 	V	S			N
	<ul style="list-style-type: none"> Format 2 USE [FOR] SCREEN PROCESSING ON data-name 	V	S			N
	<ul style="list-style-type: none"> Format 3 USE [FOR] ATTENTION-KEY PROCESSING 	V	S			N
	<ul style="list-style-type: none"> Format 4 USE [FOR] INTERRUPT PROCESSING [ON] file-name 	V	S			N
	<ul style="list-style-type: none"> Format 5 USE [FOR] FUNCTION-KEY PROCESSING 	V	S			N
WAIT	<ul style="list-style-type: none"> Format 1 WAIT <literal id [SECOND[S]]> 	V	S			N
	<ul style="list-style-type: none"> Format 2 WAIT ON <lock-variable file-name I-O> 	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
WRITE	<ul style="list-style-type: none"> • Format 1 WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [<BEFORE AFTER> [ADVANCING] <<i>id-1</i> [LINE LINES] <i>integer-1</i> PAGE>] 	V	S			N
	<ul style="list-style-type: none"> • Format 2 WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [; INVALID [KEY] <i>imperative-stmt-1</i>] 	V	S			N
	<ul style="list-style-type: none"> • Format 3 WRITE <i>record-name-1</i> [FROM ATTENTION-INDEX] 	V	S			N
	<ul style="list-style-type: none"> • Format 4 WRITE FUNCTION-INDEX 	V	S			N

1-204 Supported COBOL Statements
Supported HP COBOL II/XL statements

Supported HP COBOL II/XL statements

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S

Special registers

Registers	Parser	CE
CURRENT-DATE	V	N
DEBUG-ITEM	V	N
LINAGE-COUNTER	V	N
RETURN-CODE	V	N
TALLY	V	S
TIME-OF-DAY	V	N
WHEN-COMPILED	V	N

Literals

Literal type	Format	Parser	CE
Nonnumeric literal	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
	<ul style="list-style-type: none"> Format 2: with apostrophes Example: 'THIS ISN'T WRONG' 	V	S
Numeric literals	<ul style="list-style-type: none"> Format Example: -2.71828 	V	S
Numeric octal literals	<ul style="list-style-type: none"> Format %456 	V	S
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$99,999,999.99 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format <i>text-name-1</i> [<IN OF> <i>library-name-1</i>] 	V	S
to Procedure Division	<ul style="list-style-type: none"> Format 1 <i>paragraph-name-1</i> [<IN OF> <i>section-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2 <i>section-name-1</i> 	V	S

1-206 Supported COBOL Statements
Supported HP COBOL II/XL statements

Division	Format	Parser	CE
to Data Division	<ul style="list-style-type: none"> Format 1: simple data reference <i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> Format 2: subscripting <i>data-name-1</i> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) (<i>leftmost-character-position: [length]</i>) 	V	S
	<ul style="list-style-type: none"> Format 3 <<i>condition-name-1</i> <i>data-name-1</i>> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 4 LINAGE-COUNTER [<IN OF> <i>file-name-2</i>] 	V	S
Subscript	<ul style="list-style-type: none"> Format <<i>integer-1</i> ALL <i>data-name-3</i> [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>]> 	V	S
Reference modification	<ul style="list-style-type: none"> Format <<i>data-name-1</i> 	V	S
	FUNCTION <i>function-name-1</i> (<i>arguments</i>)>	VO	N
	(<i>leftmost-character-position: [length]</i>)	V	S

COBOL program structure

Program type	Format	Parser	CE
Single program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> END-PROGRAM <i>program-name-1</i>. 	V	S

Program type	Format	Parser	CE
Nested program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> <i>nested source program</i> END-PROGRAM <i>program-name-1</i>. 	N	N

Identification Division

Paragraphs	Format	Parser	CE
Program Identification Division	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name</i> [[IS] <COMMON [INITIAL] INITIAL [COMMON]> [PROGRAM]]. [AUTHOR. <i>comment-entry</i>] [INSTALLATION. <i>comment-entry</i>] [DATE-WRITTEN. <i>comment-entry</i>] [DATE-COMPILED. <i>comment-entry</i>] [SECURITY. <i>comment-entry</i>] [REMARKS. <i>comment-entry</i>] 	V	S

Environment Division

Input-output section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [[ORGANIZATION [IS]] SEQUENTIAL] [ACCESS [MODE] [IS] SEQUENTIAL] [[FILE] STATUS [IS] <i>stat-item</i>].	VO	N
	<ul style="list-style-type: none"> Format 2: indexed file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] INDEXED [ACCESS [MODE] [IS] <SEQUENTIAL RANDOM DYNAMIC>] RECORD [KEY] [IS] <i>data-name-2</i> [WITH DUPLICATES] [ALTERNATE RECORD [KEY] [IS] <i>data-name-3</i> [[WITH] DUPLICATES]] [[FILE] STATUS [IS] <i>stat-item</i>].	VO	N
	<ul style="list-style-type: none"> Format 3: relative file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] RELATIVE [ACCESS [MODE] [IS] <SEQUENTIAL [RELATIVE [KEY] [IS] <i>data-name-4</i>] <RANDOM DYNAMIC> RELATIVE [KEY] [IS] <i>data-name-4</i> >] [[FILE] STATUS [IS] <i>stat-item</i>].	VO	N
	<ul style="list-style-type: none"> Format 4: random-access file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] ACCESS [MODE] [IS] RANDOM ACTUAL KEY IS <i>data-name-1</i> [[FILE] STATUS [IS] <i>stat-name</i>].	VO	N
I-O-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential I-O I-O-CONTROL. [SAME [RECORD SORT SORT-MERGE] AREA FOR <i>file-name-1</i> { <i>file-name-2</i> } ...] [MULTIPLE FILE [TAPE] [CONTAINS] <i>file-name-3</i> [POSITION] <i>integer-1</i>].	VO	N

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COMPUTER paragraph	<ul style="list-style-type: none"> Format OBJECT-COMPUTER. [<i>computer-name</i> [MEMORY [SIZE] <i>integer</i> <WORDS CHARACTERS MODULES>] [(PROGRAM) [COLLATING] SEQUENCE [IS] <i>alphabet-name</i>] [SEGMENT-LIMIT [IS] <i>priority-number</i>].] 	VO	N
SOURCE-COMPUTER paragraph	<ul style="list-style-type: none"> Format SOURCE-COMPUTER. [<i>computer-name</i> [[WITH] DEBUGGING MODE].] 	VO	N
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> Format SPECIAL-NAMES. [[<<i>feature-name</i> <i>switch-name</i> <i>device-name</i>> <[IS] <i>mnemonic-name-1</i> <ON [STATUS] [IS] <i>condition-1</i> [OFF [STATUS] [IS] <i>conditional-2</i> OFF [STATUS] [IS] <i>conditional-2</i> [ON [STATUS] [IS] <i>conditional-1</i>] [IS] <i>conditional-1</i>> <ON [STATUS] [IS] <i>condition-1</i> [OFF [STATUS] [IS] <i>conditional-2</i> OFF [STATUS] [IS] <i>conditional-2</i> [ON [STATUS] [IS] <i>conditional-1</i>> > 	V	PS
	[ALPHABET <i>alphabet-name-1</i> [IS] <STANDARD-1 STANDARD-2 NATIVE EBCDIC EBCDIK <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i> >]	VO	N
	[SYMBOLIC [CHARACTERS] <i>symbolic-character</i> [ARE IS] <i>integer-1</i> [IN] <i>alphabet-name-2</i>]	VO	N
	[CLASS <i>class-name-1</i> [IS] <i>literal-4</i> [<THROUGH THRU> <i>literal-5</i>]	VO	N
	[CURRENCY [SIGN] [IS] <i>literal-6</i>]	VO	N
	[DECIMAL-POINT [IS] COMMA] [,]	V	N

Data Division

Entries and clauses	Format	Parser	CE
Data Division	<ul style="list-style-type: none"> Format: program and method DATA DIVISION DATA DIVISION. [FILE SECTION. [<i>file-description-entry</i> <i>record-description-entry</i>]] [WORKING-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [LINKAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] 	V	S

1-210 Supported COBOL Statements
Supported HP COBOL II/XL statements

Entries and clauses	Format	Parser	CE
File Description (FD) Entry	<ul style="list-style-type: none"> Format 1 FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] <i>integer-1</i> TO <i>integer-2</i> <CHARACTERS RECORDS>] [RECORDING MODE [IS] <F V U S>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [CODE-SET [IS] <i>alphabet-name</i>]. 	V	S
	<ul style="list-style-type: none"> Format 2: sort/merge files SD <i>file-name-1</i> [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i>] [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>]. 	V	S
Data Description Entry	<ul style="list-style-type: none"> Format 1 <i>level-number</i> <<i>data-name-1</i> FILLER> [<i>redefines-clause</i>] [<i>external-clause</i>] [<i>global-clause</i>] [<i>picture-clause</i>] [<i>usage-clause</i>] [<i>sign-clause</i>] [<i>occurs-clause</i>] [<i>synchronized-clause</i>] [<i>justified-clause</i>] [<i>blank-when-zero-clause</i>] [<i>value-clause</i>] 	V	S
	<ul style="list-style-type: none"> Format 2 66 <i>data-name-1</i> <i>renames-clause</i>. 	V	S
	<ul style="list-style-type: none"> Format 3 88 <i>condition-name</i> <i>value-clause</i>. 	V	S
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> Format BLANK [WHEN] <ZERO ZEROS ZEROES> 	V	S
EXTERNAL clause	<ul style="list-style-type: none"> Format [IS] EXTERNAL 	V	N
GLOBAL clause	<ul style="list-style-type: none"> Format [IS] EXTERNAL 	V	N

Entries and clauses	Format	Parser	CE
JUSTIFIED clause	<ul style="list-style-type: none"> Format <JUSTIFIED JUST> [RIGHT] 	VO	N
OCCURS clause	<ul style="list-style-type: none"> Format 1: fixed-length tables OCCURS <i>integer-2</i> [TIMES] [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2: variable-length tables OCCURS <i>integer-1</i> TO <i>integer-2</i> [TIMES] DEPENDING [ON] <i>data-name-1</i> [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
PICTURE clause	<ul style="list-style-type: none"> Format <PICTURE PIC> [IS] <i>character-string</i> 	V	S
REDEFINES clause	<ul style="list-style-type: none"> Format <i>level-number</i> <<i>data-name-1</i> FILLER> REDEFINES <i>data-name-2</i> 	V	S
RENAMES clause	<ul style="list-style-type: none"> Format 66 <i>data-name-1</i> RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>] 	V	S
SIGN clause	<ul style="list-style-type: none"> Format SIGN [IS] <LEADING TRAILING>[SEPARATE CHARACTER] 	V	N
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZED SYNC> [LEFT RIGHT] 	VO	N
USAGE clause	<ul style="list-style-type: none"> Format 1: standard USAGE [IS] <BINARY COMP COMP-3 COMPUTATIONAL COMPUTATIONAL-3 DISPLAY INDEX PACKED-DECIMAL> 	V	S
VALUE clause	<ul style="list-style-type: none"> Format 1: literal value VALUE [IS] <i>literal</i> 	V	S
	<ul style="list-style-type: none"> Format 2: condition-name value 88 <i>condition-name-1</i> <VALUE [IS] VALUES [ARE]> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]. 	V	S

1-212 Supported COBOL Statements
Supported HP COBOL II/XL statements

Procedure Division

Arithmetic operators

Binary operation	Meaning	Format	Parser	CE			
				Gen.	'Read' variables	'Written' variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

Conditional Expressions

Conditions	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1</i> [IS] [NOT] <NUMERIC ALPHABETIC ALPHABETIC-LOWER ALPHABETIC-UPPER <i>class-name</i>> 	V	S	<i>id-1</i>		S
		VO	N			N
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] <[NOT] <GREATER [THAN] LESS [THAN] EQUAL [TO] > GREATER [THAN] OR EQUAL [TO] LESS [THAN] OR EQUAL [TO]> <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>		S
Intrinsic relation condition	<ul style="list-style-type: none"> Format <i>mnemonic-name</i> [IS] <[NOT] <GREATER [THAN] LESS [THAN] EQUAL [TO] > GREATER [THAN] OR EQUAL [TO] LESS [THAN] OR EQUAL [TO]> 0 	V	S	<i>mnemonic-name</i>		S
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	<i>operand-1</i>		S
Switch-status condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT <i>condition-1</i> 	V	S	<i>condition-1</i>		S
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1</i> <AND OR> <i>condition-2</i> 	V	S	<i>condition-1</i> <i>condition-2</i>		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format <i>relation-condition</i> <AND OR> [NOT] [<i>relational-operator</i>] <i>object</i> 	V	S	<i>relation-condition</i> <i>object</i>		S

1-214 Supported COBOL Statements
Supported HP COBOL II/XL statements

Statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format PROCEDURE DIVISION [USING <i>data-name-1</i>]. 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N
PROCEDURE DIVISION structure	<ul style="list-style-type: none"> Format <i>procedure division header</i> [DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE statement [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES.] <i>section-name</i> SECTION [<i>priority-number</i>]. [<i>paragraph-name</i>. [<i>sentence</i>]] 	V	PS			N
DECLARATIVES section	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE <i>sentence</i> [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FREE] [FROM <<i>mnemonic-name</i> SYSIN CONSOLE>] [[ON] INPUT ERROR <i>imperative-stmt-1</i>] [NOT [ON] INPUT ERROR <i>imperative-stmt-2</i>] [END-ACCEPT] 	V	S		<i>id</i>	N
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE DAY DAY-OF-WEEK TIME> 	V	S		<i>id</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ADD	<ul style="list-style-type: none"> Format 1 ADD <id-1 literal> TO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <id-1 literal-1> TO <id-2 literal-1> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> id-1 TO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-2	N
ALTER	<ul style="list-style-type: none"> Format ALTER 	V	S			N
	<ul style="list-style-type: none"> procedure-name-1 TO [PROCEED TO] procedure-name-2 	VO	N			N
CALL	<ul style="list-style-type: none"> Format 1: with ON OVERFLOW CALL <id-1 literal-1> [USING <[[BY] REFERENCE] id-2 [BY] CONTENT id-3>] [[ON] OVERFLOW imperative-stmt-1] [END-CALL]] 	V	S	id-1 id-2 id-3	id-2	S
	<ul style="list-style-type: none"> Format 2: with ON EXCEPTION CALL <id-1 literal-1> [USING <[[BY] REFERENCE] id-2 [BY] CONTENT id-3>] [ON] EXCEPTION imperative-stmt-1 [NOT [ON] EXCEPTION imperative-stmt-2] [END-CALL] 	V	S	id-1 id-2 id-3	id-2	S
	<ul style="list-style-type: none"> Format 3: with GIVING CALL <id-1 [INTRINSIC] literal-1> [USING < @id-2 id-2 id-2\ literal-2\>] [GIVING id-4] [ON OVERFLOW imperative-stmt-1] [END-CALL] 	V	N			N

1-216 Supported COBOL Statements
Supported HP COBOL II/XL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
	<ul style="list-style-type: none"> Format 4: with GIVING CALL <id-1 [INTRINSIC] literal-1> [USING <\\ @id-2 id-2 \id-2 \literal-2>] [GIVING id-4] [ON] EXCEPTION imperative-stmt-1 [NOT [ON] EXCEPTION imperative-stmt-2] [END-CALL] 	V	N			N
CANCEL	<ul style="list-style-type: none"> Format CANCEL <id-1 literal-1> 	V	S			N
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE file-name-1 [<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND] [WITH] <NO REWIND LOCK>] 	V	S	file-name-1	file-name-1	N
		VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files CLOSE file-name-1 [[WITH] LOCK] 	V	S	file-name-1	file-name-1	N
		VO	N			N
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE id-1 [ROUNDED] = arithmetic-expr [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-COMPUTE] 	V	S	arithmetic-expr	id-1	S
CONTINUE	<ul style="list-style-type: none"> Format CONTINUE 	V	S			N
COPY	<ul style="list-style-type: none"> Format COPY <text-name-1 literal-1> [<OF IN> <library-name literal-2>] [SUPPRESS] [REPLACING operand-1 BY operand-2] 	V	S			N/A
DELETE	<ul style="list-style-type: none"> Format DELETE file-name-1 [RECORD] [INVALID [KEY] imperative-stmt-1] [NOT INVALID [KEY] imperative-stmt-2] [END-DELETE] 	V	S	file-name-1	file-name-1	N
DISPLAY	<ul style="list-style-type: none"> Format DISPLAY <id-1 literal-1> [UPON <mnemonic-name-1 SYSOUT CONSOLE>] [[WITH] NO ADVANCING] 	V	S	id-1 mnemonic-name-1		S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <id-1 literal-1> INTO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
ENTER	<ul style="list-style-type: none"> Format ENTER language-name [routine-name] 	VO	N			N
ENTRY	<ul style="list-style-type: none"> Format 1 ENTRY literal-1 [USING id-1] 	V	S	id-1		N

1-218 Supported COBOL Statements
Supported HP COBOL II/XL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
EVALUATE	<ul style="list-style-type: none"> Format EVALUATE <id-1 literal-1 expr-1 TRUE FALSE> [ALSO <id-2 literal-2 expr-2 TRUE FALSE>] WHEN phrase-1 [ALSO phrase-2] imperative-stmt-1 [WHEN OTHER imperative-stmt-2] [END-EVALUATE] Phrases phrase-1 and phrase-2 should be represented in the following form: <ANY condition-1 TRUE FALSE [NOT] <id-3 literal-1 arithmetic-expr-1> [<THROUGH THRU> <id-4 literal-2 arithmetic-expr-2>]> 	V	S	id-1 expr-1 id-2 expr-2 condition-1 id-3 id-4 arithmetic-expr1 arithmetic-expr2		N
EXAMINE	<ul style="list-style-type: none"> Format EXAMINE id <TALLYING <UNTIL FIRST ALL LEADING> literal-1 [REPLACING BY literal-2] REPLACING <UNTIL FIRST ALL LEADING> literal-3 BY literal-4> 	V	S	id		N
EXCLUSIVE	<ul style="list-style-type: none"> Format EXCLUSIVE file-name [CONDITIONALLY] 	V	S	file-name		N
EXIT	<ul style="list-style-type: none"> Format paragraph-name. EXIT. 	V	S			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
GOBACK	<ul style="list-style-type: none"> Format GOBACK 	V	S			N
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO [TO] procedure-name-1 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO [TO] procedure-name-1 DEPENDING [ON] id-1 	V	S	id-1		N
IF	<ul style="list-style-type: none"> Format IF condition-1 THEN <stmt-1 NEXT SENTENCE> [ELSE <stmt-2 NEXT SENTENCE>] [END-IF] 	V	S	condition-1		N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
INITIALIZE	<ul style="list-style-type: none"> Format INITIALIZE <i>id-1</i> [REPLACING <ALPHABETIC ALPHANUMERIC NUMERIC ALPHANUMERIC-EDITED NUMERIC-EDITED> [DATA] BY <<i>id-2</i> <i>literal-1</i>>] 	V	S	<i>id-2</i>	<i>id-1</i>	N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-5</i> <i>id-4</i> <i>id-3</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 4 INSPECT <i>id-1</i> CONVERTING <<i>id-6</i> <i>literal-4</i>> TO <<i>id-7</i> <i>literal-5</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] 	V	S	<i>id-1</i> <i>id-6</i> <i>id-7</i> <i>id-4</i>	<i>id-1</i>	N
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> [[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>] 	V	S	<i>file-name-1</i> <i>data-name-1</i> <i>file-name-2</i> <i>file-name-3</i>	<i>file-name-4</i>	N
	<ul style="list-style-type: none"> USING <i>file-name-2</i> <i>file-name-3</i> <OUTPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i>> 	VO	N			N
		V	S			N
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <<i>id-1</i> <i>literal-1</i>> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	N

1-220 Supported COBOL Statements
Supported HP COBOL II/XL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <id-1 literal-1> BY id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-MULTIPLY] 	V	S	id-1 id-2	id-3	S
OPEN	<ul style="list-style-type: none"> Format 1: sequential files OPEN <INPUT file-name-1 [REVERSED WITH NO REWIND] OUTPUT file-name-2 [WITH NO REWIND] I-O file-name-3 EXTEND file-name-4> 	V	S	file-name-1 file-name-2 file-name-3 file-name-4	file-name-1 file-name-2 file-name-3 file-name-4	N
		VO	N			N
		V	S			N
		VO	N			N
		V	S			N
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM <procedure-name-1 [<THROUGH THRU> procedure-name-2] imperative-stmt-1 END-PERFORM> 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM <procedure-name-1 [<THROUGH THRU> procedure-name-2] <id-1 integer-1> TIMES imperative-stmt-1 END-PERFORM> 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase and END-PERFORM PERFORM [procedure-name-1 [<THROUGH THRU> procedure-name-2]] [[WITH] TEST <BEFORE AFTER>] UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	condition-1		N
	<ul style="list-style-type: none"> Format 4: out-of-line PERFORM ... VARYING PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 AFTER <id-5 index-name-3> FROM <id-6 index-name-4 literal-3> BY <id-7 literal-4> UNTIL condition-2 	N	N			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
	<ul style="list-style-type: none"> Format 5: in-line PERFORM ... VARYING PERFORM [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	id-3 id-4 condition-1	id-2	N
READ	<ul style="list-style-type: none"> Format 1: all files READ file-name-1 [NEXT] [RECORD] 	V	S	file-name-1	file-name-1 id-1	N
		VO	N			N
	[[INTO id-1] [[AT] END imperative-stmt-1] [NOT [AT] END imperative-stmt-2] [END-READ]	V	S			S
	<ul style="list-style-type: none"> Format 2: indexed files READ file-name-1 RECORD [INTO id-1] [KEY [IS] data-name-1] [INVALID [KEY] imperative-stmt-3] [NOT INVALID [KEY] imperative-stmt-4] [END-READ] 	V	S	file-name-1 data-name-1	file-name-1 id-1	N
RELEASE	<ul style="list-style-type: none"> Format RELEASE record-name-1 [FROM id-1] 	V	S	record-name-1 id-1	record-name-1	N
REPLACE	<ul style="list-style-type: none"> Format 1 REPLACE ==pseudo-text-1== BY ==pseudo-text-2==. 	V	S			N/ A
	<ul style="list-style-type: none"> Format 2 REPLACE OFF. 	V	S			N/ A
RETURN	<ul style="list-style-type: none"> Format RETURN file-name-1 [RECORD] [INTO id-1] [[AT] END imperative-stmt-1] [NOT [AT] END imperative-stmt-2] [END-RETURN] 	V	S	file-name-1	id-1	N
REWRITE	<ul style="list-style-type: none"> Format REWRITE record-name-1 [FROM id-1] [INVALID [KEY] imperative-stmt-1] [NOT INVALID [KEY] imperative-stmt-2] [END-REWRITE] 	V	S	file-name-1 id-1	file-name-1	N

1-222 Supported COBOL Statements
Supported HP COBOL II/XL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [[AT] END <i>imperative-stmt-1</i>] WHEN <i>condition-1</i> <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>condition-1</i>	<i>id-2</i>	N
	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [[AT] END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> [IS] EQUAL [TO] <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> [IS] EQUAL [TO] <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>data-name-1</i>		N
SEEK	<ul style="list-style-type: none"> Format SEEK <i>file-name</i> [RECORD] 	N	N			N
SET	<ul style="list-style-type: none"> Format 1: basic table handling SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-2</i> <i>integer-1</i>> 	V	S	<i>index-name-2</i> <i>id-2</i>	<i>index-name-1</i> <i>id-1</i>	N
	<ul style="list-style-type: none"> Format 2: adjusting indexes SET <i>index-name-3</i> <UP BY DOWN BY> <<i>id-3</i> <i>integer-2</i>> 	V	S	<i>index-name-3</i> <i>id-3</i>	<i>index-name-3</i>	N
	<ul style="list-style-type: none"> Format 3: external switches SET <i>mnemonic-name-1</i> TO <ON OFF> 	V	S		<i>mnemonic-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: condition-names SET <i>condition-name-1</i> TO TRUE 	V	S		<i>condition-variable</i>	S
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> [[WITH] DUPLICATES [IN] [ORDER]] 	V	S	<i>file-name-1</i> <i>data-name-1</i> <i>file-name-2</i>	<i>file-name-3</i>	N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N			N
	<USING <i>file-name-2</i> INPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i> > <GIVING <i>file-name-3</i> OUTPUT PROCEDURE [IS] <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i> >	V	S			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [KEY [IS] <EQUAL [TO] = LESS [THAN] < GREATER [THAN] > NOT LESS [THAN] NOT < NOT GREATER [THAN] NOT > LESS [THAN] OR EQUAL [TO] <= GREATER [THAN] OR EQUAL [TO] >= > <i>data-name-1</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-START] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N
STRING	<ul style="list-style-type: none"> Format STRING <<i>id-1</i> <i>literal-1</i>> DELIMITED [BY] <<i>id-2</i> <i>literal-2</i> SIZE> INTO <i>id-3</i> [[WITH] POINTER <i>id-4</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-STRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>	<i>id-3</i> <i>id-4</i>	S
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <<i>id-1</i> <i>literal-1</i>> FROM <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <<i>id-1</i> <i>literal-1</i>> FROM <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> <i>id-1</i> FROM <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-SUBTRACT] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N
UN-EXCLUSIVE	<ul style="list-style-type: none"> Format UN-EXCLUSIVE <i>file-name</i> 	V	S	<i>file-name</i>		N

1-224 Supported COBOL Statements
Supported HP COBOL II/XL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING <i>id-1</i> [DELIMITED [BY] [ALL] <<i>id-2</i> <i>literal-1</i>> [OR [ALL] <<i>id-3</i> <i>literal-2</i>>]] INTO <i>id-4</i> [DELIMITER [IN] <i>id-5</i>] [COUNT [IN] <i>id-6</i>] [[WITH] POINTER <i>id-7</i>] [TALLYING [IN] <i>id-8</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-UNSTRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-7</i> <i>id-8</i>	<i>id-4</i> <i>id-5</i> <i>id-6</i> <i>id-7</i> <i>id-8</i>	S
USE	<ul style="list-style-type: none"> Format 1: EXCEPTION ERROR declarative USE [GLOBAL] AFTER [STANDARD] <EXCEPTION ERROR> PROCEDURE [ON] <<i>file-name-1</i> INPUT OUTPUT I-O EXTEND> 	VO	N			N
	<ul style="list-style-type: none"> Format 2: LABEL declarative USE [GLOBAL] AFTER [STANDARD] BEGINNING [FILE] LABEL PROCEDURE [ON] <<i>file-name-1</i> INPUT OUTPUT I-O EXTEND> 	VO	N			N
	<ul style="list-style-type: none"> Format 3: debugging USE [FOR] DEBUGGING [ON] <<i>procedure-name</i> ALL PROCEDURES> 	V	S			N
WRITE	<ul style="list-style-type: none"> Format 1: sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [<BEFORE AFTER> [ADVANCING] <<<i>id-2</i> <i>integer-1</i>> [LINE LINES] PAGE <i>mnemonic-name</i>>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i> <i>id-2</i> <i>mnemonic-name</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: relative, indexed or random-access files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N

Intrinsic functions

Functions	Parser	CE	
		Gen.	DB
ACOS	V	S	N
ANNUITY	V	S	N
ASIN	V	S	N
ATAN	V	S	N
CHAR	V	S	N
COS	V	S	N
CURRENT-DATE	V	S	N
DATE-OF-INTEGER	V	S	N
DAY-OF-INTEGER	V	S	N
FACTORIAL	V	S	N
INTEGER	V	S	N
INTEGER-OF-DATE	V	S	N
INTEGER-OF-DAY	V	S	N
INTEGER-PART	V	S	N
LENGTH	V	S	N
LOG	V	S	N
LOG10	V	S	N
LOWER-CASE	V	S	N
MAX	V	S	N
MEAN	V	S	N
MEDIAN	V	S	N

1-226 Supported COBOL Statements
Supported HP COBOL II/XL statements

Functions	Parser	CE	
		Gen.	DB
MIDRANGE	V	S	N
MIN	V	S	N
MOD	V	S	N
NUMVAL	V	S	N
NUMVAL-C	V	S	N
ORD	V	S	N
ORD-MAX	V	S	N
ORD-MIN	V	S	N
PRESENT-VALUE	V	S	N
RANDOM	V	S	N
RANGE	V	S	N
REM	V	S	N
REVERSE	V	S	N
SIN	V	S	N
SQRT	V	S	N
STANDARD-DEVIATION	V	S	N
SUM	V	S	N
TAN	V	S	N
UPPER-CASE	V	S	N
VARIANCE	V	S	N
WHEN-COMPILED	V	S	N

Preprocessor commands

Commands	Parser	CE
\$COMMENT	V	S
\$CONTROL	V	S
\$COPYRIGHT	V	S
\$DEFINE	V	S
\$EDIT	V	S
\$IF	V	S
\$INCLUDE	V	S
\$PAGE	V	S
\$PREPROCESSOR	V	S
\$SET	V	S
\$TITLE	V	S
\$VERSION	V	S

1-228 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Supported Fujitsu COBOL85 (M Series) statements

Note: Object-oriented COBOL statements are not supported.

COBOL language structure

Figurative constants

Constants	Parser	CE
[ALL] <ZERO ZEROS ZEROES>	V	S
[ALL] <SPACE SPACES>	V	S
[ALL] <HIGH-VALUE HIGH-VALUES>	V	S
[ALL] <LOW-VALUE LOW-VALUES>	V	S
[ALL] NULL	V	S
[ALL] <QUOTE QUOTES>	V	S
ALL literal	V	S
[ALL] symbolic-character	V	N

Special registers

Registers	Parser	CE
AQLCA	N	N
EDIT-COLOR	N	N
EDIT-CURSOR	N	N
EDIT-MODE	N	N
EDIT-OPTION	N	N

Registers	Parser	CE
EDIT-STATUS	N	N
FCOM	V	N
LINAGE-COUNTER	V	N
LINE-COUNTER	V	N
PAGE-COUNTER	V	N
PROGRAM-STATUS	V	N
RETURN-CODE	V	N
SHIFT OUT / SHIFT IN	V	N
SORT-CONTROL	V	N
SORT-CORE-SIZE	V	N
SORT-FILE-SIZE	V	N
SORT-MESSAGE	V	N
SORT-MODE-SIZE	V	N
SORT-RETURN	V	N
SORT-STATUS	V	N
TALLY	V	N
WHEN COMPILED	V	N

Literals

Literal type	Format	Parser	CE
Nonnumeric literals	<ul style="list-style-type: none"> Format Example: "THIS ISN'T WRONG" 	V	S

1-230 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Literal type	Format	Parser	CE
Numeric literals	<ul style="list-style-type: none"> • Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> • Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
Hexadecimal nonnumeric literals	<ul style="list-style-type: none"> • Format X"AB123FE9236" 	V	S
National literals	<ul style="list-style-type: none"> • Format N"national-string" NC"national-string" NN"national-string" NA"national-string" NK"national-string" NH"national-string" G"national-string" NX"AB123FE9236" 	V	S
Boolean literals	<ul style="list-style-type: none"> • Format B"010010111010" 	V	S
PICTURE character string	<ul style="list-style-type: none"> • Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
Qualification	<ul style="list-style-type: none"> Format 1 <data-name-1 condition-name-1 index-name-1> <<IN OF> data-name-2 ... [<IN OF> file-name-1] <IN OF file-name-1>> 	V	N
	<ul style="list-style-type: none"> Format 2 paragraph-name-1 <IN OF> section-name-1 	V	N
	<ul style="list-style-type: none"> Format 3 text-name-1 <IN OF> library-name-1 	V	N
	<ul style="list-style-type: none"> Format 4 LINAGE-COUNTER <IN OF> file-name-1 	V	N
	<ul style="list-style-type: none"> Format 5 <PAGE-COUNTER LINE-COUNTER> <IN OF> report-name-1 	V	N
	<ul style="list-style-type: none"> Format 6 data-name-1 <<IN OF> data-name-2 [<IN OF> report-name-1] <IN OF> report-name-1> 	V	N
	<ul style="list-style-type: none"> Format 7 <EDIT-MODE EDIT-OPTION EDIT-COLOR EDIT-STATUS EDIT-CURSOR> <IN OF> id-1 	N	N
Subscripting	<ul style="list-style-type: none"> Format <condition-name-1 data-name-1> (<ALL integer-1 data-name-2 [<+ -> integer-2] index-name-1> [<+ -> integer-3] ...) 	V	S
Function-identifier	<ul style="list-style-type: none"> Format FUNCTION function-name-1 [(argument-1 ...)] 	V	S
Reference modifier	<ul style="list-style-type: none"> Format <data-name-1 (leftmost-character-position: [length])> 	V	S
Identifier	<ul style="list-style-type: none"> Format 1 function-identifier-1 	V	S
	<ul style="list-style-type: none"> Format 2 data-name-1 [<IN OF> data-name-2] ... [<IN OF> <file-name-1 report-name-1>] [(subscript ...)] [reference-modifier] 	V	S

COBOL program structure

1-232 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Program type	Format	Parser	CE
Single program	<ul style="list-style-type: none"> Format IDENTIFICATION DIVISION. PROGRAM-ID. <i>program-description-entry id-division-entries</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> END-PROGRAM <i>program-name-1</i>. 	V	S

Identification Division

Paragraphs	Format	Parser	CE
Program Identification Division	<ul style="list-style-type: none"> Format IDENTIFICATION DIVISION. PROGRAM-ID. <i>program-name</i>. [AUTHOR. [<i>comment-entry</i>]] [INSTALLATION. [<i>comment-entry</i>]] [DATE-WRITTEN. [<i>comment-entry</i>]] [DATE-COMPILED. [<i>comment-entry</i>]] [SECURITY. [<i>comment-entry</i>]] 	V	S

Environment Division

Input-output section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN [TO] < <i>file-id-1</i> <i>file-id-literal-1</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [[ORGANIZATION [IS]] SEQUENTIAL] [ACCESS [MODE] [IS] SEQUENTIAL] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-3</i>]] [FORMAT [IS] <i>data-name-4</i>] [GROUP [IS] <i>data-name-5</i>] [PADDING [CHARACTER] [IS] < <i>data-name-6</i> <i>literal-1</i> >] [RECORD DELIMITER [IS] STANDARD-1]	VO	N
	<ul style="list-style-type: none"> Format 2: indexed file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN [TO] < <i>file-id-1</i> <i>file-id-literal-1</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] INDEXED [ACCESS [MODE] [IS] <SEQUENTIAL RANDOM DYNAMIC>] RECORD [KEY] [IS] <i>data-name-2</i> [[WITH] DUPLICATES] [ALTERNATE RECORD [KEY] [IS] <i>data-name-3</i> [[WITH] DUPLICATES]] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N

1-234 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Paragraphs and entries	Format	Parser	CE
	<ul style="list-style-type: none"> Format 3: relative file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN [TO] < <i>file-id-1</i> <i>file-id-literal-1</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] RELATIVE [ACCESS [MODE] [IS] <SEQUENTIAL [RELATIVE [KEY] [IS] <i>data-name-4</i>] <RANDOM DYNAMIC> RELATIVE [KEY] [IS] <i>data-name-4</i> >] [[FILE] STATUS [IS] <i>data-name-2</i> [<i>data-name-3</i>]].	VO	N
	<ul style="list-style-type: none"> Format 4: presentation file-control entries FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN [TO] <i>file-id-1</i>	V	S
	[ORGANIZATION [IS]] SEQUENTIAL [ACCESS [MODE] [IS] SEQUENTIAL] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-2</i>]] [FORMAT [IS] <i>data-name-3</i>] [GROUP [IS] <i>data-name-4</i>] [[SYMBOLIC] DESTINATION [IS] < <i>data-name-5</i> <i>literal-1</i> >] [DESTINATION-1 [IS] <i>data-name-6</i>] [DESTINATION-2 [IS] <i>data-name-7</i>] [DESTINATION-3 [IS] <i>data-name-8</i>] [PROCESSING [MODE] [IS] <i>data-name-9</i>] [[SELECTED] FUNCTION [IS] <i>data-name-10</i>] [UNIT CONTROL [IS] <i>data-name-11</i>] [MESSAGE MODE [IS] <i>data-name-12</i>] [END KEY [IS] <i>data-name-13</i>] [SESSION CONTROL [IS] <i>data-name-14</i>] [PROCESSING CONTROL [IS] <i>data-name-15</i>] [MESSAGE CLASS [IS] <i>data-name-16</i>] [PROCESSING TIME [IS] <i>data-name-17</i>] [MESSAGE CODE [IS] <i>data-name-18</i>] [MESSAGE OWNER [IS] <i>data-name-19</i>]	VO	N
	<ul style="list-style-type: none"> Format 5: direct file-control entries FILE-CONTROL. SELECT <i>file-name-1</i> ASSIGN [TO] <i>file-id-1</i> [[FOR] MULTIPLE UNIT]	V	S
	[ACCESS [MODE] [IS] <SEQUENTIAL RANDOM>] [ACTUAL KEY [IS] <i>data-name-1</i>] [<FILE-LIMIT [IS] FILE-LIMITS [ARE]> < <i>data-name-1</i> <i>literal-1</i> > <THROUGH THRU> < <i>data-name-2</i> <i>literal-2</i> >] [PROCESSING [MODE] [IS] SEQUENTIAL] [RESERVE <NO <i>integer-1</i> >] [ALTERNATE] [AREA AREAS]] [TRACK-LIMIT [IS] <i>integer-1</i> [TRACK TRACKS]]	VO	N

Paragraphs and entries	Format	Parser	CE
I-O-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential I-O I-O-CONTROL. <RERUN ON <i>file-id-1</i> [EVERY] <<i>integer-1</i> RECORDS END [OF] <REEL UNIT>> [OF] <i>file-name-1</i> SAME [RECORD] [AREA] [FOR] <i>file-name-3 file-name-4</i> MULTIPLE FILE [TAPE] [CONTAINS] <i>file-name-5</i> [POSITION] <i>integer-2</i>>. 	VO	N
	<ul style="list-style-type: none"> Format 2: relative and indexed I-O I-O-CONTROL. <RERUN ON <i>file-id-1</i> [EVERY] <i>integer-1</i> RECORDS [OF] <i>file-name-1</i> SAME [RECORD] [AREA] [FOR] <i>file-name-3 file-name-4</i>> 	VO	N
	<ul style="list-style-type: none"> Format 3: sort-merge I-O I-O-CONTROL. [SAME <RECORD SORT SORT-MERGE> [AREA] [FOR] <i>file-name-3 file-name-4</i>.] 	VO	N
	<ul style="list-style-type: none"> Format 4: presentation file I-O I-O-CONTROL. [SAME [RECORD] [AREA] [FOR] <i>file-name-3 file-name-4</i>] [APPLY <MULTICONVERSATION-MODE MULTICON>] [APPLY SAVED-AREA [TO] <i>data-name-1</i>] 	VO	N

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COMPUTER paragraph	<ul style="list-style-type: none"> Format OBJECT-COMPUTER. [<i>computer-name</i> [MEMORY [SIZE] <i>integer</i> <WORDS CHARACTERS MODULES>] [[PROGRAM] [COLLATING] SEQUENCE [IS] <i>alphabet-name</i>].] 	VO	N
SOURCE-COMPUTER paragraph	<ul style="list-style-type: none"> Format SOURCE-COMPUTER. [<i>computer-name</i> [[WITH] DEBUGGING MODE].] 	VO	N

1-236 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Paragraphs and entries	Format	Parser	CE
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> Format SPECIAL-NAMES. [<i>environment-name-1</i> [IS] <i>mnemonic-name-1</i> <i>environment-name-2</i> <[IS] <i>mnemonic-name-2</i> <i>entry-1</i> <i>entry-1</i> >] where <i>entry-1</i> is: <i>entry-1</i> : <ON [STATUS] [IS] <i>condition-1</i> [OFF [STATUS] [IS] <i>conditional-2</i>] OFF [STATUS] [IS] <i>condition-2</i> [ON [STATUS] [IS] <i>conditional-1</i>]>	V	N
	[ALPHABET <i>alphabet-name-1</i> [IS] <STANDARD-1 STANDARD-2 NATIVE <i>function-name</i> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i> ALSO <i>literal-3</i>]>]	VO	N
	[SYMBOLIC [CHARACTERS] <i>symbolic-character</i> [ARE IS] <i>integer-1</i> [[IN] <i>alphabet-name-2</i>] [SYMBOLIC CONSTANT <i>symbolic-constant-1</i> IS <i>literal-8</i> ...] ...]	VO	N
	[CLASS <i>class-name-1</i> [IS] <i>literal-4</i> [<THROUGH THRU> <i>literal-5</i>]]	VO	N
	[CURRENCY [SIGN] [IS] <i>literal-6</i> [[WITH] PICTURE SYMBOL <i>literal-7</i>]]	VO	N
	[DECIMAL-POINT [IS] COMMA].	V	

Data Division

Entries and clauses	Format	Parser	CE
Data Division	<ul style="list-style-type: none"> Format DATA DIVISION. [BASED-STORAGE SECTION. [<i>77-level-description-entry</i> <i>record-description-entry</i>] ...] [FILE SECTION. [<i>file-description-entry</i> <i>record-description-entry</i> <i>sort/merge-file-description-entry</i> <i>record-description-entry</i> <i>report-file-description-entry</i>]] [WORKING-STORAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [CONSTANT SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [LINKAGE SECTION. [<i>record-description-entry</i> <i>data-item-description-entry</i>]] [REPORT SECTION. [<i>report-description-entry</i> <i>report-group-description-entry</i>]]	V	S

Entries and clauses	Format	Parser	CE
File Description (FD) Entry	<ul style="list-style-type: none"> • Format 1: sequential files FD <i>file-name-1</i> [[IS] EXTERNAL [IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <[CHARACTERS] RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i> [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED <i>data-name-2</i>>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [CODE-SET [IS] <i>alphabet-name</i>] [CONTROL <RECORD [IS] RECORDS [ARE]> <i>data-name-9</i>]. 	V	S
	<ul style="list-style-type: none"> • Format 2: relative and indexed files FD <i>file-name-1</i> [[IS] EXTERNAL [IS] GLOBAL] [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <[CHARACTERS] RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i> [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED [VALUE OF <i>data-name-2</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>]. 	V	S
	<ul style="list-style-type: none"> • Format 3: sort/merge files SD <i>file-name-1</i> [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i> [CHARACTERS] [DEPENDING [ON] <i>data-name-1</i>]>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] 	V	S
	<ul style="list-style-type: none"> • Format 4: report files FD <i>file-name-1</i> [BLOCK [CONTAINS] [<i>integer-1</i> TO] <i>integer-2</i> <[CHARACTERS] RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [CHARACTERS] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [CHARACTERS]>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED <i>data-name-2</i>>] [VALUE OF <i>system-name-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [CODE-SET [IS] <i>alphabet-name</i>] [<REPORT [IS] REPORTS [ARE]> <i>report-name-1</i>]. 	V	S

1-238 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Entries and clauses	Format	Parser	CE
Data Description Entry	<ul style="list-style-type: none"> • Format 1 <i>level-number <data-name-1 FILLER></i> <i>[redefines-clause] [picture-clause] [usage-clause] [sign-clause] [occurs-clause] [synchronized-clause] [justified-clause] [blank-when-zero-clause] [value-clause] [character-type-clause] [printing-position-clause]</i> 	V	S
	<ul style="list-style-type: none"> • Format 2 <i>66 data-name-1 renames-clause.</i> 	V	S
	<ul style="list-style-type: none"> • Format 3 <i>88 condition-name value-clause.</i> 	V	S
Report Description Entry	<ul style="list-style-type: none"> • Format <i>RD report-name-1</i> <i>[CODE literal-1] [<CONTROL [IS] CONTROLS [ARE]> <data-name-1 FINAL data-name-1>]</i> <i>[PAGE [LIMIT [IS] LIMITS [ARE]] integer-1 [LINE LINES]] [HEADING integer-2] [FIRST DETAIL integer-3]</i> <i>[LAST DETAIL integer-4] [FOOTING integer-5]</i> 	V	S

Entries and clauses	Format	Parser	CE
Report Group Description	<ul style="list-style-type: none"> • Format 1 01 [<i>data-name-1</i>] [LINE [NUMBER] [IS] <<i>integer-1</i> [[ON] NEXT PAGE] PLUS <i>integer-2</i>>] [NEXT GROUP [IS] <<i>integer-3</i> PLUS <i>integer-4</i> NEXT PAGE>] TYPE [IS] <REPORT HEADING RH PAGE HEADING PH <CONTROL HEADING CH> <<i>data-name-2</i> FINAL> DETAIL DE <CONTROL FOOTING CF> <<i>data-name-3</i> FINAL> PAGE FOOTING PF REPORT FOOTING RF> [[USAGE [IS]] DISPLAY] [CHARACTER TYPE [IS] <<MODE-1 MODE-2 MODE-3> [BY] <i>mnemonic-name-1</i> <i>mnemonic-name-2</i>>] 	V	N
	<ul style="list-style-type: none"> • Format 2 <i>level-number</i> [<i>data-name-1</i>] [LINE NUMBER IS <<i>integer-1</i> [[ON] NEXT PAGE] PLUS <i>integer-2</i>>] [[USAGE [IS]] DISPLAY] [CHARACTER TYPE [IS] <<MODE-1 MODE-2 MODE-3> [BY] <i>mnemonic-name-1</i> <i>mnemonic-name-2</i>>] 	V	N
	<ul style="list-style-type: none"> • Format 3 <i>level-number</i> [<i>data-name-1</i>] <PICTURE PIC> IS <i>character-string</i> [[USAGE [IS]] DISPLAY] [[SIGN [IS]] <LEADING TRAILING> SEPARATE [CHARACTER]] [<JUSTIFIED JUST> RIGHT] [BLANK WHEN ZERO] [LINE [NUMBER] [IS] <<i>integer-1</i> [[ON] NEXT PAGE] PLUS <i>integer-2</i>>] [COLUMN NUMBER IS <i>integer-3</i>] [[PRINTING] POSITION [IS] <i>integer-4</i>] SOURCE IS <i>id-1</i> <VALUE IS <i>literal</i> SUM <i>id-2</i> [UPON <i>data-name-2</i>]> [RESET ON <<i>data-name-3</i> FINAL>] [GROUP INDICATE] [CHARACTER TYPE [IS] <<MODE-1 MODE-2 MODE-3> [BY] <i>mnemonic-name-1</i> <i>mnemonic-name-2</i>>] 	V	N
BLANK WHEN ZERO clause	<ul style="list-style-type: none"> • Format BLANK [WHEN] <ZERO ZEROS ZEROES> 	V	N
CHARACTER TYPE clause	<ul style="list-style-type: none"> • Format <[CHARACTER TYPE [IS]] <MODE-1 MODE-2 MODE-3> [[BY] <i>mnemonic-1</i>] CHARACTER TYPE [IS] <i>mnemonic-2</i> CHARACTER TYPE [IS] <<i>printing-mode-name-1</i> <i>printing-mode-name-2</i> ... DEPENDING [ON] <i>data-name-5</i>> [OR <<i>printing-mode-name-3</i> <i>printing-mode-name-4</i> ... DEPENDING [ON] <i>data-name-5</i>>] 	VO	N
JUSTIFIED clause	<ul style="list-style-type: none"> • Format <JUSTIFIED JUST> [RIGHT] 	VO	N

1-240 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Entries and clauses	Format	Parser	CE
OCCURS clause	<ul style="list-style-type: none"> • Format 1: fixed-length tables OCCURS <i>integer-2</i> [TIMES] [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
	<ul style="list-style-type: none"> • Format 2: variable-length tables OCCURS <i>integer-1</i> TO <i>integer-2</i> [TIMES] DEPENDING [ON] <i>data-name-1</i> [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
PICTURE clause	<ul style="list-style-type: none"> • Format <PICTURE PIC> [IS] <i>character-string</i> 	V	S
PRINTING POSITION clause	<ul style="list-style-type: none"> • Format [PRINTING] POSITION [IS] <i>integer</i> [[BY] <i>positioning-unti-name-1</i>] 	V	S
REDEFINES clause	<ul style="list-style-type: none"> • Format <i>level-number</i> <<i>data-name-1</i> FILLER> REDEFINES <i>data-name-2</i> 	V	S
RENAMES clause	<ul style="list-style-type: none"> • Format <i>66 data-name-1</i> RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>] 	V	S
SIGN clause	<ul style="list-style-type: none"> • Format SIGN [IS] <LEADING TRAILING>[SEPARATE CHARACTER] 	V	S
SYNCHRONIZED clause	<ul style="list-style-type: none"> • Format <SYNCHRONIZED SYNC> [LEFT RIGHT] 	VO	N
USAGE clause	<ul style="list-style-type: none"> • Format 1: standard USAGE [IS] <BINARY BIT COMP COMP-5 COMPUTATIONAL COMPUTATIONAL-5 DISPLAY INDEX PACKED-DECIMAL POINTER> 	V	S
VALUE clause	<ul style="list-style-type: none"> • Format 1: literal value VALUE [IS] <i>literal</i> 	V	S
	<ul style="list-style-type: none"> • Format 2: condition-name value <i>88 condition-name-1</i> <VALUE [IS] VALUES [ARE]> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]. 	V	S

Procedure Division

Arithmetic operators

Operation	Meaning	Format	Parser	CE			
				Gen.	'Read' variables	'Written' variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

1-242 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Conditional Expressions

Conditions	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1</i> [IS] [NOT] <NUMERIC ALPHABETIC ALPHABETIC-LOWER ALPHABETIC-UPPER BOOLEAN KANJI DBCS <i>class-name</i>> 	V	S	<i>id-1</i>		S
		VO	N			N
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] <[NOT] <GREATER [THAN] LESS [THAN] EQUAL [TO] > < => GREATER [THAN] OR EQUAL [TO] LESS [THAN] OR EQUAL [TO] >= <= > <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>		S
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] [NOT] <POSITIVE NEGATIVE ZERO> 	V	S	<i>operand-1</i>		N
Switch-status condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Negated simple condition	<ul style="list-style-type: none"> Format NOT <i>condition-1</i> 	V	S	<i>condition-1</i>		S
Combined condition	<ul style="list-style-type: none"> Format <i>condition-1</i> <AND OR> <i>condition-2</i> 	V	S	<i>condition-1</i> <i>condition-2</i>		S
Abbreviated combined relation condition	<ul style="list-style-type: none"> Format <i>relation-condition</i> <AND OR> [NOT] [<i>relational-operator</i>] <i>object</i> 	V	S	<i>object</i>		S

Statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION structure	<ul style="list-style-type: none"> Format 1 PROCEDURE DIVISION [USING <i>data-name-1</i> ...]. [DECLARATIVES section] section-name SECTION [paragraph-name. [sentence ...] ...] 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N
	<ul style="list-style-type: none"> Format 2 PROCEDURE DIVISION [USING <i>data-name-1</i> ...]. paragraph-name. [sentence ...] ... 	V	S	<i>data-name-1</i>	<i>data-name-1</i>	N
DECLARATIVES section	<ul style="list-style-type: none"> Format DECLARATIVES. section-name SECTION. USE [paragraph-name [section ...] ...] ... END-DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FROM <mnemonic-name>] 	V	S	<i>id</i>		N
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE DAY DAY-OF-WEEK TIME> 	V	S	<i>id</i>		N
	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> FROM <CONSOLE SYSIN> 	V	S	<i>id</i>		N

1-244 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
ADD	<ul style="list-style-type: none"> Format 1 ADD <id-1 literal> TO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <id-1 literal-1> TO <id-2 literal-1> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> id-1 TO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-ADD] 	V	S	id-1 id-2	id-2	S
ALTER	<ul style="list-style-type: none"> Format ALTER 	V	S			N
	<ul style="list-style-type: none"> procedure-name-1 TO [PROCEED TO] procedure-name-2 	VO	N			N
CALL	<ul style="list-style-type: none"> Format 1 CALL <id-1 literal-1> [USING <[[BY] REFERENCE] id-2 [BY] CONTENT <id-3 literal-2>>] [ON] OVERFLOW imperative-stmt-1] [END-CALL] 	V	S	id-1 id-2 id-3	id-2	S
	<ul style="list-style-type: none"> Format 2 CALL <id-1 literal-1> [USING <[[BY] REFERENCE] id-2 [BY] CONTENT <id-3 literal-2>>] [[ON] EXCEPTION imperative-stmt-1] [NOT [ON] EXCEPTION imperative-stmt-2] [END-CALL] 	VO	N			N
CANCEL	<ul style="list-style-type: none"> Format CANCEL <id-1 literal-1> 	V	S	id-1		N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	[<REEL UNIT> [[FOR] REMOVAL] [WITH] <NO REWIND LOCK>]	VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files CLOSE <i>file-name-1</i> 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
	[[WITH] LOCK]	VO	N			N
COMPUTE	<ul style="list-style-type: none"> Format 1 COMPUTE <i>id-1</i> [ROUNDED] = <i>arithmetic-expr</i> [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-COMPUTE] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S
	<ul style="list-style-type: none"> Format 2 COMPUTE <i>id-1</i> = <i>boolean-expr</i> 	V	S		<i>id-1</i>	S
CONTINUE	<ul style="list-style-type: none"> Format CONTINUE 	V	S			N
DELETE	<ul style="list-style-type: none"> Format DELETE <i>file-name-1</i> [RECORD] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-DELETE] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISPLAY	<ul style="list-style-type: none"> Format DISPLAY <<i>id-1</i> <i>literal-1</i>> [UPON <<i>mnemonic-name-1</i> >] [[WITH] NO ADVANCING] 	V	S	<i>id-1</i>		S

1-246 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <id-1 literal-1> INTO id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
ENTER	<ul style="list-style-type: none"> Format ENTER language-name-1 [procedure-name-1] 	V	N			N
ENTRY	<ul style="list-style-type: none"> Format ENTRY literal-1 [USING id-1] 	V	S		id-1	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
EVALUATE	<ul style="list-style-type: none"> Format EVALUATE <id-1 literal-1 expr-1 TRUE FALSE> [ALSO <id-2 literal-2 expr-2 TRUE FALSE>] WHEN phrase-1 [ALSO phrase-2] imperative-stmt-1 [WHEN OTHER imperative-stmt-2] [END-EVALUATE] Phrases phrase-1 and phrase-2 should be represented in the following form: <ANY condition-1 TRUE FALSE [NOT] <id-3 literal-1 arithmetic-expr-1> [<THROUGH THRU> <id-4 literal-2 arithmetic-expr-2>]> 	V	S	id-1 id-2 id-3 id-4		
EXAMINE	<ul style="list-style-type: none"> Format EXAMINE id <TALLYING <UNTIL FIRST ALL LEADING> literal-1 [REPLACING BY literal-2] REPLACING <UNTIL FIRST ALL LEADING> literal-3 BY literal-4> 	V	S	id	id	
EXIT	<ul style="list-style-type: none"> Format EXIT. 	V	S			N
EXIT PERFORM	<ul style="list-style-type: none"> Format EXIT [[TO] TEST [OF]] PERFORM. 	V	S			
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
GENERATE	<ul style="list-style-type: none"> Format GENERATE <data-name-1 report-name-1> 	V	S	data-name-1		
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO [TO] procedure-name-1 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO [TO] procedure-name-1 DEPENDING [ON] id-1 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3: MORE-LABELS GO [TO] MORE-LABELS 	V	S			N

1-248 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
IF	<ul style="list-style-type: none"> Format IF <i>condition-1</i> THEN <<i>stmt-1</i> NEXT SENTENCE> <ELSE <i>stmt-2</i> [END-IF] ELSE NEXT SENTENCE END-IF> 	V	S	<i>condition-1</i>		N
INITIALIZE	<ul style="list-style-type: none"> Format INITIALIZE <i>id-1</i> [REPLACING <ALPHABETIC ALPHANUMERIC NUMERIC ALPHANUMERIC-EDITED NUMERIC-EDITED NATIONAL NATIONAL-EDITED BOOLEAN EGCS> [DATA] BY <<i>id-2</i> <i>literal-1</i>>] 	V	S	<i>id-1</i>	<i>id-2</i>	N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-3</i> <i>id-4</i> <i>id-5</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 4 INSPECT <i>id-1</i> CONVERTING <<i>id-6</i> <i>literal-4</i>> TO <<i>id-7</i> <i>literal-5</i>> [BEFORE AFTER] [INITIAL] <<i>id-4</i> <i>literal-2</i>> 	V	S	<i>id-1</i> <i>id-4</i> <i>id-6</i> <i>id-7</i>	<i>id-1</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> 	V	S			N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N			N
	<ul style="list-style-type: none"> USING <i>file-name-2 file-name-3</i> <OUTPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i>> 	V	S			N
MOVE	<ul style="list-style-type: none"> Format 1 MOVE <<i>id-1</i> <i>literal-1</i>> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	S
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY <<i>id-1</i> <i>literal-1</i>> BY <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-MULTIPLY] 	V	S	<i>id-1</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <<i>id-1</i> <i>literal-1</i>> BY <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-MULTIPLY] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S

1-250 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
OPEN	<ul style="list-style-type: none"> Format 1: sequential files OPEN <INPUT <i>file-name-1</i> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N
	[REVERSED WITH NO REWIND]	VO	N			
	OUTPUT <i>file-name-2</i>	V	S			
	[WITH NO REWIND]	VO	N			
	I-O <i>file-name-3</i> EXTEND <i>file-name-4</i> >	V	S			
	<ul style="list-style-type: none"> Format 2: indexed and relative files OPEN <INPUT <i>file-name-1</i> OUTPUT <i>file-name-2</i> I-O <i>file-name-3</i> EXTEND <i>file-name-4</i>> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM [<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] [<i>imperative-stmt-1</i> END-PERFORM] 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM [<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <<i>id-1</i> <i>integer-1</i>> TIMES [<i>imperative-stmt-1</i> END-PERFORM] 	V	S	<i>id-1</i>		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase PERFORM [<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] [[WITH] TEST <BEFORE AFTER>] UNTIL <i>condition-1</i> [<i>imperative-stmt-1</i> END-PERFORM] 	V	S	<i>condition-1</i>		N
	<ul style="list-style-type: none"> Format 4: with VARYING phrase PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] [[WITH] TEST <BEFORE AFTER>] VARYING <<i>id-2</i> <i>index-name-1</i>> FROM <<i>id-3</i> <i>index-name-2</i> <i>literal-1</i>> BY <<i>id-4</i> <i>literal-2</i>> UNTIL <i>condition-1</i> [AFTER <<i>id-5</i> <i>index-name-3</i>> FROM <<i>id-6</i> <i>index-name-4</i> <i>literal-3</i>> BY <<i>id-7</i> <i>literal-4</i>> UNTIL <i>condition-2</i>] [<i>imperative-stmt-1</i> END-PERFORM] 	V	S	<i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i> <i>id-7</i>	<i>id-2</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
READ	<ul style="list-style-type: none"> Format 1: sequential retrieval READ <i>file-name-1</i>	V	S			N
	[NEXT] [RECORD]	VO	N		<i>id-1</i>	N
	[INTO <i>id-1</i>] [[AT] END <i>imperative-stmt-1</i>] [NOT [AT] END <i>imperative-stmt-2</i>] [END-READ]	V	S			
	<ul style="list-style-type: none"> Format 2: random retrieval READ <i>file-name-1</i> [RECORD] [INTO <i>id-1</i>] [KEY [IS] <i>data-name-1</i>] [INVALID [KEY] <i>imperative-stmt-3</i>] [NOT INVALID [KEY] <i>imperative-stmt-4</i>] [END-READ]	V	S		<i>id-1</i>	N
RELEASE	<ul style="list-style-type: none"> Format RELEASE <i>record-name-1</i> [FROM <i>id-1</i>]	V	S	<i>id-1</i>		N
RETURN	<ul style="list-style-type: none"> Format RETURN <i>file-name-1</i> [RECORD] [INTO <i>id-1</i>] [[AT] END <i>imperative-stmt-1</i>] [NOT [AT] END <i>imperative-stmt-2</i>] [END-RETURN]	V	S		<i>id-1</i>	N
REWRITE	<ul style="list-style-type: none"> Format REWRITE <i>rec-name-1</i> [FROM <i>id-1</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-REWRITE]	V	S	<i>rec-name-1</i> <i>id-1</i>	<i>rec-name-1</i>	N

1-252 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <i><id-2 index-name-1></i>][[AT] END <i>imperative-stmt-1</i>] WHEN <i>condition-1 <imperative-stmt-2 NEXT SENTENCE></i> [END-SEARCH] 	V	S	<i>id-1</i> <i>condition-1</i>		N
	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [[AT] END <i>imperative-stmt-1</i>] WHEN <i><data-name-1 [IS] EQUAL [TO] <id-3 literal-1 arithmetic-expr-1> condition-name-1></i> [AND <i><data-name-2 [IS] EQUAL [TO] <id-4 literal-2 arithmetic-expr-2> condition-name-2></i>] <i><imperative-stmt-2 NEXT SENTENCE></i> [END-SEARCH] 	V	S	<i>id-1</i> <i>id-3</i> <i>id-4</i>		N
SEEK	<ul style="list-style-type: none"> Format SEEK <i>file-name-1</i> [RECORD] 	N	N			N
SET	<ul style="list-style-type: none"> Format 1: basic table handling SET <i><index-name-1 id-1></i> TO <i><index-name-2 id-2 integer-1></i> 	V	S	<i>id-2</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 2: adjusting indexes SET <i>index-name-3</i> <UP BY DOWN BY> <i><id-3 integer-2></i> 	V	S	<i>index-name-3</i>	<i>id-3</i>	N
	<ul style="list-style-type: none"> Format 3: external switches SET <i>mnemonic-name-1</i> TO <ON OFF> 	V	S		<i>mnemonic-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: condition-names SET <i>condition-name-1</i> TO TRUE 	V	S		<i>condition-name-1</i>	N
	<ul style="list-style-type: none"> Format 5: pointer data items SET <i><id-4 ADDRESS OF id-5></i> TO <i><id-6 ADDRESS OF id-7 literal-1></i> 	V	S	<i>id-6</i>	<i>id-4</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [KEY] <i>data-name-1</i> [[WITH] DUPLICATES [IN] [ORDER]] 	V	S			N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N			N
	<USING <i>file-name-2</i> INPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>]> <GIVING <i>file-name-3</i> OUTPUT PROCEDURE [IS] <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i>]>	V	S			N
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [KEY [IS] <EQUAL [TO] = LESS [THAN] < GREATER [THAN] > NOT LESS [THAN] NOT < NOT GREATER [THAN] NOT > LESS [THAN] OR EQUAL [TO] <= GREATER [THAN] OR EQUAL [TO] >= > <i>data-name-1</i>] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-START] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N
STRING	<ul style="list-style-type: none"> Format STRING <<i>id-1</i> <i>literal-1</i>> DELIMITED [BY] <<i>id-2</i> <i>literal-2</i> SIZE> INTO <i>id-3</i> [[WITH] POINTER <i>id-4</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-STRING] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i> <i>id-4</i>	N

1-254 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <id-1 literal-1> FROM id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <id-1 literal-1> FROM <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> id-1 FROM id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	N
SUPPRESS	<ul style="list-style-type: none"> Format SUPPRESS [PRINTING] 	V	S			N
TERMINATE	<ul style="list-style-type: none"> Format TERMINATE report-name-1 	V	S			N
TRANSACTION	<ul style="list-style-type: none"> Format TRANSACTION <START END CANCEL NOTE id-1> 	V	S			N
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING id-1 [DELIMITED [BY] [ALL] <id-2 literal-1> [OR [ALL] <id-3 literal-2>]] INTO id-4 [DELIMITER [IN] id-5] [COUNT [IN] id-6] [[WITH] POINTER id-7] [TALLYING [IN] id-8] [[ON] OVERFLOW imperative-stmt-1] [NOT [ON] OVERFLOW imperative-stmt-2] [END-UNSTRING] 	V	S	id-1 id-2 id-3 id-5	id-4 id-6 id-7 id-8	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
WRITE	<ul style="list-style-type: none"> Format 1: sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i> [<BEFORE AFTER> [ADVANCING] <<<i>id-1</i> <i>integer-1</i>> [LINE LINES] <i>mnemonic-name-1</i> PAGE>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [END-WRITE] 	VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files WRITE <i>record-name-1</i> [FROM <i>id-1</i> [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-WRITE] 	V	S	<i>id-1</i>	<i>record-name-1</i>	N

Intrinsic functions

Functions	Parser	CE
ADDR	V	N
LENG	V	N

Compile-directing statements and directives

Statements and directives	Format	Parser	CE
*CONTROL (*CBL)	<ul style="list-style-type: none"> Format <*CONTROL *CBL> <SOURCE NOSOURCE LIST NOLIST MAP NOMAP> 	VO	N
BASIS	<ul style="list-style-type: none"> Format [<i>sequence-number</i>] BASIS <<i>basis-name</i> <i>literal-1</i>> 	VO	N

1-256 Supported COBOL Statements
Supported Fujitsu COBOL85 (M Series) statements

Statements and directives	Format	Parser	CE
COPY	<ul style="list-style-type: none"> Format 1 COPY <text-name-1 literal-1> [<OF IN> library-name] [REPLACING operand-1 BY operand-2] 	V	S
	<ul style="list-style-type: none"> Format 2 COPY <text-name-1 literal-1> [<OF IN> library-name] DISJOINING word-3 JOINING word-4 AS <PREFIX SUFFIX> 	V	S
DELETE	<ul style="list-style-type: none"> Format [sequence-number] DELETE sequence-number-field 	VO	N
EJECT	<ul style="list-style-type: none"> Format EJECT 	VO	N
INSERT	<ul style="list-style-type: none"> Format [sequence-number] INSERT sequence-number-field 	VO	N
INCLUDE	<ul style="list-style-type: none"> Format INCLUDE <text-name-1 literal-1> [<OF IN> library-name] 	V	S
NOTE	<ul style="list-style-type: none"> Format NOTE character-string 	V	S
REPLACE	<ul style="list-style-type: none"> Format 1 REPLACE ==pseudo-text-1== BY ==pseudo-text-2==. 	V	S
	<ul style="list-style-type: none"> Format 2 REPLACE OFF. 	V	S
SKIP1/2/3	<ul style="list-style-type: none"> Format <SKIP1 SKIP2 SKIP3>. 	VO	N
TITLE	<ul style="list-style-type: none"> Format TITLE <i>literal</i>. 	VO	N

Statements and directives	Format	Parser	CE
USE	<ul style="list-style-type: none"> • Format 1 USE [GLOBAL] AFTER [STANDARD] <EXCEPTION ERROR> PROCEDURE [ON] <file-name-1 INPUT OUTPUT I-O EXTEND> 	V	S
	<ul style="list-style-type: none"> • Format 2 USE BEFORE REPORTING <i>id-1</i> 	VO	N
	<ul style="list-style-type: none"> • Format 3 USE FOR DEAD-LOCK 	V	N

Supported AIM Network Database Statements

Clauses	Format	Parser	CE
CONNECT	<ul style="list-style-type: none"> Format CONNECT [<i>literal-1</i> <i>id-1</i>] TO <ALL <<i>literal-2</i> <i>id-2</i>> [NEXT PRIOR <i>integer</i> AT <i>id-3</i>> 	V	N
DISCONNECT	<ul style="list-style-type: none"> Format DISCONNECT [<i>literal-1</i> <i>id-1</i>] FROM <ALL [<i>literal-2</i> <i>id-2</i>> 	V	N
ERASE	<ul style="list-style-type: none"> Format ERASE <<i>literal-1</i> <i>id-1</i>> [SELECTIVE ALL] 	V	N
FIND	<ul style="list-style-type: none"> Format FIND <i>record-selection-expression</i> [[AT] END GO [TO] <i>procedure-name-1</i>] 	V	N
FINISH	<ul style="list-style-type: none"> Format FINISH [<i>literal-1</i>] 	V	N
GET	<ul style="list-style-type: none"> Format GET <i>record-selection-expression</i> [[AT] END GO [TO] <i>procedure-name-1</i>] 	V	N
IF	<ul style="list-style-type: none"> Format IF DB-EXCEPTION <IS [NOT] <= < > >> <i>integer-1</i> [<AND OR> [DB-EXCEPTION] <IS [NOT] <= < > >> <i>integer-2</i> ...] <GO TO <i>procedure-name-1</i> PERFORM <i>procedure-name-2</i> [THRU <i>procedure-name-3</i>] CALL <i>literal-1</i> USING <i>literal-2</i>> [END-IF] 	V	N
MODIFY	<ul style="list-style-type: none"> Format MODIFY <<i>literal-1</i> <i>id-1</i>> 	V	N
READY	<ul style="list-style-type: none"> Format READY [<i>schema-name-1</i>] 	V	N
STORE	<ul style="list-style-type: none"> Format 1 STORE <<i>literal-1</i> <i>id-1</i>> [TO <ALL <<i>literal-2</i> <i>id-2</i>> [NEXT PRIOR <i>integer</i> AT <i>id-3</i>>] 	V	N
	<ul style="list-style-type: none"> Format 2 STORE <<i>literal-1</i> <i>id-1</i>> TO <<i>literal-2</i> <i>id-2</i>> [AT] END GO [TO] <i>procedure-name-1</i> 	V	N

Clauses	Format	Parser	CE
USE	<ul style="list-style-type: none">• Format USE FOR DB-EXCEPTION	V	N

1-260 Supported COBOL Statements
Supported SIEMENS COBOL statements

Supported SIEMENS COBOL statements

Note: Object-oriented COBOL statements are not supported.

COBOL language structure

Figurative constants

Constants	Parser	CE
ZERO / ZEROS / ZEROES	V	S
SPACE / SPACES	V	S
HIGH-VALUE / HIGH-VALUES	V	S
LOW-VALUE / LOW-VALUES	V	S
QUOTE / QUOTES	V	S
ALL literal	V	S
NULL / NULLS	V	S

Special registers

Registers	Parser	CE
ADDRESS OF	V	PS
DEBUG ITEM	V	N
LENGTH OF	V	S
LINAGE-COUNTER	V	N
RETURN-CODE	V	N
SHIFT OUT / SHIFT IN	V	N
SORT-CONTROL	V	N
SORT-CORE-SIZE	V	N
SORT-FILE-SIZE	V	N
SORT-MESSAGE	V	N
SORT-MODE-SIZE	V	N
SORT-RETURN	V	N
TALLY	V	S
WHEN COMPILED	V	N
SORT-EOW	V	N
CBL-CTR	V	N

1-262 Supported COBOL Statements
Supported SIEMENS COBOL statements

Literals

Literal type	Format	Parser	CE
Nonnumeric literal	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
	<ul style="list-style-type: none"> Format 2: with apostrophes Example: 'THIS ISN'T WRONG' 	V	S
	<ul style="list-style-type: none"> Format 3: with double-byte characters "EBCDIC-data<D1D2>EBCDIC-data" 	N	N
	<ul style="list-style-type: none"> Format 4: hexadecimal notation X"hexadecimal-digits" 	V	S
	<ul style="list-style-type: none"> Format 5: null-terminated Z"dddd" 	VO	N
Numeric literals	<ul style="list-style-type: none"> Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
DBCS literals	<ul style="list-style-type: none"> Format 1: G"<D1D2D3>" 	V	S
	<ul style="list-style-type: none"> Format 2: N"<D1D2D3>" 	N	N
PICTURE character string	<ul style="list-style-type: none"> Format Example: \$9(5).9(2)CR 	V	S

Referencing names

Division	Format	Parser	CE
to COPY libraries	<ul style="list-style-type: none"> Format <i>text-name-1</i> [<IN OF> <i>library-name-1</i>] 	V	S
to Procedure Division	<ul style="list-style-type: none"> Format 1: <i>paragraph-name-1</i> [<IN OF> <i>section-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2: <i>section-name-1</i> 	V	S
to Data Division	<ul style="list-style-type: none"> Format 1: simple data reference <i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> Format 2: subscripting <i>data-name-1</i> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) (<i>leftmost-character-position: [length]</i>) 	V	S
	<ul style="list-style-type: none"> Format 3: <<i>condition-name-1</i> <i>data-name-1</i>> [<IN OF> <i>data-name-2</i>] [<IN OF> <i>file-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 4: LINAGE-COUNTER [<IN OF> <i>file-name-2</i>] 	V	S
Condition names	<ul style="list-style-type: none"> Format 1: Data Division <i>condition-name-1</i> [<IN OF> <i>data-name-1</i>] [<IN OF> <i>file-name-1</i>] (<i>subscript</i>) 	V	S
	<ul style="list-style-type: none"> Format 2: Special-Names paragraph <i>condition-name-1</i> [<IN OF> <i>mnemonic-name-1</i>] 	V	S
Subscript	<ul style="list-style-type: none"> Format <<i>integer-1</i> <i>arithmetic-expr</i> ALL <i>data-name-3</i> [<+ -> <i>integer-2</i>] <i>index-name-1</i> [<+ -> <i>integer-3</i>> 	V	S
Reference modification	<ul style="list-style-type: none"> Format <<i>data-name-1</i> 	V	S
	<ul style="list-style-type: none"> FUNCTION <i>function-name-1</i> (<i>arguments</i>)> 	VO	N
	<ul style="list-style-type: none"> (<i>leftmost-character-position: [length]</i>) 	V	S

1-264 Supported COBOL Statements
Supported SIEMENS COBOL statements

COBOL program structure

Program type	Format	Parser	CE
Single program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> END-PROGRAM <i>program-name-1</i> .	V	S
Nested program	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name-1</i> ENVIRONMENT DIVISION. <i>env-division-content</i> DATA DIVISION. <i>data-division-content</i> PROCEDURE DIVISION. <i>proc-division-content</i> <i>nested source program</i> END-PROGRAM <i>program-name-1</i> .	N	N

Identification Division

Paragraphs	Format	Parser	CE
Program Identification Division	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. PROGRAM-ID. <i>program-name</i> [[IS] <RECURSIVE COMMON [INITIAL] INITIAL [COMMON]> [PROGRAM]]. [AUTHOR. [<i>comment-entry</i>]] [INSTALLATION. [<i>comment-entry</i>]] [DATE-WRITTEN. [<i>comment-entry</i>]] [DATE-COMPILED. [<i>comment-entry</i>]] [SECURITY. [<i>comment-entry</i>]]	V	S
Class Identification Division	<ul style="list-style-type: none"> Format <IDENTIFICATION ID> DIVISION. CLASS-ID. <i>class-name-1</i> INHERITS <i>class-name-2</i> [METAClass [IS] <i>class-name-3</i>]. [AUTHOR. [<i>comment-entry</i>]] [INSTALLATION. [<i>comment-entry</i>]] [DATE-WRITTEN. [<i>comment-entry</i>]] [DATE-COMPILED. [<i>comment-entry</i>]] [SECURITY. [<i>comment-entry</i>]]	N	N

Paragraphs	Format	Parser	CE
Method Identification Division	<ul style="list-style-type: none"> • Format <IDENTIFICATION ID> DIVISION. METHOD-ID. <i>method-name-1</i> [[IS] [METHOD] OVERRIDE]. [AUTHOR. <i>comment-entry</i>] [INSTALLATION. <i>comment-entry</i>] [DATE-WRITTEN. <i>comment-entry</i>] [DATE-COMPILED. <i>comment-entry</i>] [SECURITY. <i>comment-entry</i>] 	N	N

Environment Division

Input-output section

Paragraphs and entries	Format	Parser	CE
FILE-CONTROL paragraph	<ul style="list-style-type: none"> Format 1: sequential file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [[ORGANIZATION [IS]] SEQUENTIAL] [PADDING [CHARACTER] [IS] < <i>data-name-5</i> <i>literal-2</i> >] [RECORD DELIMITER [IS] <STANDARD-1 <i>assignment-name-2</i> >] [ACCESS [MODE] [IS] SEQUENTIAL] [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 2: indexed file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] INDEXED [ACCESS [MODE] [IS] <SEQUENTIAL RANDOM DYNAMIC>] RECORD [KEY] [IS] <i>data-name-2</i> [PASSWORD [IS] <i>data-name-6</i>] ALTERNATE RECORD [KEY] [IS] <i>data-name-3</i> [[WITH] DUPLICATES] [PASSWORD [IS] <i>data-name-7</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 3: relative file-control entries FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[RESERVE <i>integer</i> [AREA AREAS]] [ORGANIZATION [IS]] RELATIVE [ACCESS [MODE] [IS] <SEQUENTIAL [RELATIVE [KEY] [IS] <i>data-name-4</i>] <RANDOM DYNAMIC> RELATIVE [KEY] [IS] <i>data-name-4</i> >] [PASSWORD [IS] <i>data-name-6</i>] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N
	<ul style="list-style-type: none"> Format 4: line-sequential file-control entries (all platforms except VM) FILE-CONTROL. SELECT [OPTIONAL] <i>file-name-1</i> ASSIGN <[TO] <i>assignment-name-1</i> USING <i>data-name-9</i> >	V	S
	[ORGANIZATION [IS]] LINE SEQUENTIAL [ACCESS [MODE] [IS] SEQUENTIAL] [[FILE] STATUS [IS] <i>data-name-1</i> [<i>data-name-8</i>]].	VO	N

Paragraphs and entries	Format	Parser	CE
I-O-CONTROL paragraph	<ul style="list-style-type: none"> • Format 1: sequential I-O I-O-CONTROL. <RERUN ON <assignment-name-1 file-name-1> [EVERY] <integer-1 RECORDS END [OF] <REEL UNIT>> [OF] file-name-1 SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4 MULTIPLE FILE [TAPE] [CONTAINS] file-name-5 [POSITION] integer-2 APPLY WRITE-ONLY [ON] file-name-2>. 	VO	N
	<ul style="list-style-type: none"> • Format 2: relative and indexed I-O I-O-CONTROL. <RERUN ON <assignment-name-1 file-name-1> [EVERY] integer-1 RECORDS [OF] file-name-1 SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4> 	VO	N
	<ul style="list-style-type: none"> • Format 3: line-sequential I-O I-O-CONTROL. SAME [RECORD] [AREA] [FOR] file-name-3 file-name-4. 	VO	N
	<ul style="list-style-type: none"> • Format 4: sort-merge I-O I-O-CONTROL. [RERUN [ON] <file-name-1 assignment-name-1> [[EVERY] SORT [OF] <file-name-2 assignment-name-2>]] SAME <RECORD SORT SORT-MERGE> [AREA] [FOR] file-name-3 file-name-4. 	VO	N

1-268 Supported COBOL Statements
Supported SIEMENS COBOL statements

Configuration section

Paragraphs and entries	Format	Parser	CE
OBJECT-COMPUTER paragraph	<ul style="list-style-type: none"> Format OBJECT-COMPUTER. [<computer-name <id id> <id literal > [MEMORY [SIZE] integer <WORDS CHARACTERS MODULES>] [[PROGRAM] [COLLATING] SEQUENCE [IS] alphabet-name] [SEGMENT-LIMIT [IS] priority-number].] 	VO	Y
REPOSITORY paragraph	<ul style="list-style-type: none"> Format REPOSITORY. [CLASS class-name-1 [[IS] external-class-name-1]]. 	N	N
SOURCE-COMPUTER paragraph	<ul style="list-style-type: none"> Format SOURCE-COMPUTER. [<computer-name id id id literal > [[WITH] DEBUGGING MODE].] 	VO	Y
SPECIAL-NAMES paragraph	<ul style="list-style-type: none"> Format SPECIAL-NAMES. [environment-name-1 [IS] mnemonic-name-1 environment-name-2 <[IS] mnemonic-name-2 entry-1 entry-1>] where entry-1 is: entry-1: <ON [STATUS] [IS] condition-1 [OFF [STATUS] [IS] conditional-2] OFF [STATUS] [IS] condition-2 [ON [STATUS] [IS] conditional-1]> ARGUMENT-NUMBER [IS] mnemonic-name-3 ARGUMENT-VALUE [IS] mnemonic-name-4 ENVIRONMENT-NAME [IS] mnemonic-name-5 ENVIRONMENT-VALUE [IS] mnemonic-name-6 	V	PS
	[ALPHABET alphabet-name-1 [IS] <STANDARD-1 STANDARD-2 NATIVE EBCDIC literal-1 [<THROUGH THRU> literal-2 ALSO literal-3>]	VO	N
	[SYMBOLIC [CHARACTERS] symbolic-character [ARE IS] integer-1 [IN] alphabet-name-2]	VO	N
	[CLASS class-name-1 [IS] literal-4 [<THROUGH THRU> literal-5]]	VO	N
	[CURRENCY [SIGN] [IS] literal-6 [[WITH] PICTURE SYMBOL literal-7]]	VO	N
	[DECIMAL-POINT [IS] COMMA] [.]	V	N

Data Division

Entries and clauses	Format	Parser	CE
Data Division	<ul style="list-style-type: none"> Format: program and method DATA DIVISION <p>DATA DIVISION. [FILE SECTION. <i>[file-description-entry record-description-entry]</i>] [WORKING-STORAGE SECTION. <i>[record-description-entry data-item-description-entry]</i>] [LOCAL-STORAGE SECTION. <i>[record-description-entry data-item-description-entry]</i>] [LINKAGE SECTION. <i>[record-description-entry data-item-description-entry]</i>]</p>	V	S
Data Division	<ul style="list-style-type: none"> Format: class DATA DIVISION <p>[WORKING-STORAGE SECTION. <i>[record-description-entry data-item-description-entry]</i>]</p>	N	N
File Description (FD) Entry	<ul style="list-style-type: none"> Format 1: sequential files <p>FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] <i>[integer-1 TO] integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [<CHARACTER CHARACTERS>] [CONTAINS] <i>integer-4 TO integer-5</i> [<CHARACTER CHARACTERS>] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i>] [<CHARACTER CHARACTERS>] [DEPENDING [ON] <i>data-name-1</i>]]> [LABEL <RECORD [IS] RECORDS [ARE]]> <STANDARD OMITTED <i>data-name-2</i>> [VALUE OF <i>ID-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]]> <i>data-name-4</i> [LINAGE [IS] <<i>data-name-5</i> <i>integer-8</i>> [LINES] [[WITH] FOOTING [AT] <<i>data-name-6</i> <i>integer-9</i>>] [[LINES] [AT] TOP <<i>data-name-7</i> <i>integer-10</i>>] [[LINES] [AT] BOTTOM <<i>data-name-8</i> <i>integer-11</i>>]] [RECORDING [MODE] [IS] <i>mode</i>] [CODE-SET [IS] <i>alphabet-name</i>].</p>	V	S
	<ul style="list-style-type: none"> Format 2: relative and indexed files <p>FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] <i>[integer-1 TO] integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [<CHARACTER CHARACTERS>] [CONTAINS] <i>integer-4 TO integer-5</i> [<CHARACTER CHARACTERS>] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i>] [<CHARACTER CHARACTERS>] [DEPENDING [ON] <i>data-name-1</i>]]> [LABEL <RECORD [IS] RECORDS [ARE]]> <STANDARD OMITTED [VALUE OF <i>ID-1</i> [IS] <<i>data-name-3</i> <i>literal-1</i>>] [DATA <RECORD [IS] RECORDS [ARE]]> <i>data-name-4</i>.]</p>	V	S
	<ul style="list-style-type: none"> Format 3: line-sequential files <p>FD <i>file-name-1</i> [[IS] EXTERNAL] [[IS] GLOBAL] [BLOCK [CONTAINS] <i>[integer-1 TO] integer-2</i> <CHARACTERS RECORDS>] [RECORD <[CONTAINS] <i>integer-3</i> [<CHARACTER CHARACTERS>] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i>] [<CHARACTER CHARACTERS>] [DEPENDING [ON] <i>data-name-1</i>]]>].</p>	V	S

1-270 Supported COBOL Statements
Supported *SIEMENS COBOL* statements

Entries and clauses	Format	Parser	CE
	<ul style="list-style-type: none"> • Format 4: sort/merge files • Format 4: sort/merge files SD <i>file-name-1</i> [RECORD <[CONTAINS] <i>integer-3</i> [<CHARACTER CHARACTERS>] [CONTAINS] <i>integer-4</i> TO <i>integer-5</i> [<CHARACTER CHARACTERS>] [IS] VARYING [IN] [SIZE] [[FROM] <i>integer-6</i> [TO] <i>integer-7</i>] [<CHARACTER CHARACTERS>] [DEPENDING [ON] <i>data-name-1</i>]>] [DATA <RECORD [IS] RECORDS [ARE]> <i>data-name-4</i>] [LABEL <RECORD [IS] RECORDS [ARE]> <STANDARD OMITTED <i>data-name-2</i>>] [RECORDING [MODE] [IS] <i>mode</i> 	V	S
Data Description Entry	<ul style="list-style-type: none"> • Format 1 <i>level-number</i> <<i>data-name-1</i> FILLER> [redefines-clause] [blank-when-zeros-clause] [dynamic-clause] [external-clause] [global-clause] [justified-clause] [occurs-clause] [picture-clause] [sign-clause] [synchronized-clause] [usage-clause] [value-clause] [data-format-clause] [dynamic-clause] 	V	S
	<ul style="list-style-type: none"> • Format 2 66 <i>data-name-1</i> renames-clause. 	V	S
	<ul style="list-style-type: none"> • Format 3 88 <i>condition-name</i> value-clause. 	V	S
BLANK WHEN ZEROS/ZEROES clause	<ul style="list-style-type: none"> • Format 3 BLANK [WHEN] <ZERO ZEROS ZEROES> 	V	S
DATE FORMAT clause	<ul style="list-style-type: none"> • Format DATE FORMAT [IS] <i>date-pattern</i> 	VO	N
EXTERNAL clause	<ul style="list-style-type: none"> • Format: [IS] EXTERNAL 	V	N
DYNAMIC clause	<ul style="list-style-type: none"> • Format: [IS] DYNAMIC 	V	N
GLOBAL clause		V	N
JUSTIFIED clause	<ul style="list-style-type: none"> • Format <JUSTIFIED JUST> [RIGHT] 	VO	N

Entries and clauses	Format	Parser	CE
OCCURS clause	<ul style="list-style-type: none"> Format 1: fixed-length tables OCCURS <i>integer-2</i> [TIMES] [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
	<ul style="list-style-type: none"> Format 2: variable-length tables OCCURS <i>integer-1</i> TO <i>integer-2</i> [TIMES] DEPENDING [ON] <i>data-name-1</i> [<ASCENDING DESCENDING> [KEY] [IS] <i>data-name-2</i>] [INDEXED [BY] <i>index-name-1</i>] 	V	S
PICTURE clause	<ul style="list-style-type: none"> Format <PICTURE PIC> [IS] <i>character-string</i> 	V	S
REDEFINES clause	<ul style="list-style-type: none"> Format <i>level-number</i> <<i>data-name-1</i> FILLER> REDEFINES <i>data-name-2</i> 	V	S
RENA-MES clause	<ul style="list-style-type: none"> Format 66 <i>data-name-1</i> RENAMES <i>data-name-2</i> [<THROUGH THRU> <i>data-name-3</i>] 	V	S
SIGN clause	<ul style="list-style-type: none"> Format SIGN [IS] <LEADING TRAILING>[SEPARATE CHARACTER] 	V	N
SYNCHRONIZED clause	<ul style="list-style-type: none"> Format <SYNCHRONIZED SYNC> [LEFT RIGHT] 	VO	N
USAGE clause	<ul style="list-style-type: none"> Format 1: standard USAGE [IS] <BINARY COMP COMP-1 COMP-2 COMP-3 COMP-4 COMP-5 COMPUTATIONAL COMPUTATIONAL-1 COMPUTATIONAL-2 COMPUTATIONAL-3 COMPUTATIONAL-4 COMPUTATIONAL-5 DISPLAY DISPLAY-1 INDEX PACKED-DECIMAL POINTER > 	V	S
	<ul style="list-style-type: none"> Format 2: with 'NATIVE' USAGE [IS] <BINARY NATIVE COMP COMP-1 NATIVE COMP-2 NATIVE COMP-3 COMP-4 NATIVE COMP-5 COMPUTATIONAL COMPUTATIONAL-1 NATIVE COMPUTATIONAL-2 NATIVE COMPUTATIONAL-3 COMPUTATIONAL-4 NATIVE COMPUTATIONAL-5 DISPLAY NATIVE DISPLAY-1 NATIVE INDEX PACKED-DECIMAL POINTER PROCEDURE-POINTER> 	VO	N
	<ul style="list-style-type: none"> Format 3: PROCEDURE-POINTER USAGE [IS] PROCEDURE-POINTER 	VO	N
	<ul style="list-style-type: none"> Format 4: OBJECT REFERENCE USAGE [IS] OBJECT REFERENCE [[METACLASS [OF]] <i>class-name-1</i>] 	N	N

1-272 Supported COBOL Statements
Supported SIEMENS COBOL statements

Entries and clauses	Format	Parser	CE
VALUE clause	<ul style="list-style-type: none"> • Format 1: literal value <VALUE VALUES> [FROM <i>subscript-1</i>] [<IS ARE>] <i>literal-1</i> [REPEATED <<i>integer-1</i> TIMES [TO] END>] 	V	S
	<ul style="list-style-type: none"> • Format 2: condition-name value 88 <i>condition-name-1</i> <VALUE [IS] VALUES [ARE]> <i>literal-1</i> [<THROUGH THRU> <i>literal-2</i>]. 88 <i>condition-name-2</i> <[when] set to false> 	V	S
	<ul style="list-style-type: none"> • Format 4: NULL value VALUE [IS] <NULL NULLS> 	V	S

Procedure Division

Arithmetic operators

Binary operation	Meaning	Format	Parser	CE			
				Gen.	'Read' variables	'Written' variables	DB
+	Addition	<ul style="list-style-type: none"> Format <i>oper-1 + oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
-	Subtraction	<ul style="list-style-type: none"> Format <i>oper-1 - oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
*	Multiplication	<ul style="list-style-type: none"> Format <i>oper-1 * oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
/	Division	<ul style="list-style-type: none"> Format <i>oper-1 / oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
**	Exponentiation	<ul style="list-style-type: none"> Format <i>oper-1 ** oper-2</i> 	V	S	<i>oper-1</i> <i>oper-2</i>		S
+	Multiplication by +1	<ul style="list-style-type: none"> Format <i>+ oper</i> 	V	S	<i>oper</i>		S
-	Multiplication by -1	<ul style="list-style-type: none"> Format <i>- oper</i> 	V	S	<i>oper</i>		S

1-274 Supported COBOL Statements
Supported SIEMENS COBOL statements

Conditional Expressions

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
Class condition	<ul style="list-style-type: none"> Format <i>id-1</i> [IS] [NOT] <NUMERIC ALPHABETIC ALPHABETIC-LOWER ALPHABETIC-UPPER <i>class-name</i> DBCS KANJI> 	V	S	<i>id-1</i>		S
		V O	N			N
Condition-name condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Relation condition	<ul style="list-style-type: none"> Format 1 <i>operand-1</i> [IS] <[NOT] <GREATER [THAN] LESS [THAN] EQUAL [TO] > GREATER [THAN] OR EQUAL [TO] LESS [THAN] OR EQUAL [TO]> <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>		S
	<ul style="list-style-type: none"> Format 2: pointer data items <ADDRESS OF <i>id-1</i> <i>id-2</i> NULL NULLS> [IS] [NOT] EQUAL [TO] <ADDRESS OF <i>id-3</i> <i>id-4</i> NULL NULLS> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>		N
	<ul style="list-style-type: none"> Format 3: procedure-pointer data items [<i>id-2</i> NULL NULLS] [IS] [NOT] EQUAL [TO] [<i>id-4</i> NULL NULLS] 	V	S	<i>id-2</i> <i>id-4</i>		N
	<ul style="list-style-type: none"> Format 4: object reference data items <<i>object-reference-id-1</i> SELF NULL NULLS> [IS] [NOT] EQUAL [TO] <<i>object-reference-id-2</i> SELF NULL NULLS> 	N	N			N
Sign condition	<ul style="list-style-type: none"> Format <i>operand-1</i> [IS] [NOT] <POSITIVE NEGATIVE ZEROS ZEROES> 	V	S	<i>operand-1</i>		S
Switch-status condition	<ul style="list-style-type: none"> Format <i>condition-name</i> 	V	S	<i>condition-name</i>		S
Negated simple conditions	<ul style="list-style-type: none"> Format NOT <i>condition-1</i> 	V	S	<i>condition-1</i>		S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
Combined conditions	<ul style="list-style-type: none"> Format <i>condition-1 <AND OR> condition-2</i> 	V	S	<i>condition-1 condition-2</i>		S
Abbreviated combined relation conditions	<ul style="list-style-type: none"> Format <i>relation-condition <AND OR> [NOT] [relational-operator] object</i> 	V	S	<i>relation-condition object</i>		S

Statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION header	<ul style="list-style-type: none"> Format 1: programs and methods (with BY REFERENCE) PROCEDURE DIVISION [USING [[BY] REFERENCE] <i>data-name-1</i>] [RETURNING <i>data-name-2</i>]. 	V	S	<i>data-name-1</i>	<i>data-name-1 data-name-2</i>	N
	<ul style="list-style-type: none"> Format 2: programs and methods (with BY VALUE) PROCEDURE DIVISION [USING [[BY] VALUE] <i>data-name-1</i>] [RETURNING <i>data-name-2</i>]. 	VO	N			N
	<ul style="list-style-type: none"> Format 3: classes PROCEDURE DIVISION. 	N	N			N

1-276 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PROCEDURE DIVISION structure	<ul style="list-style-type: none"> Format 1: program and method procedure division header [DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES.] <i>section-name</i> SECTION [<i>priority-number</i>]. [<i>paragraph-name</i>. [<i>sentence</i>]] 	V	PS			N
	<ul style="list-style-type: none"> Format 2: classes PROCEDURE DIVISION. [<i>method-definition</i>] 	N	N			N
DECLARATIVES section	<ul style="list-style-type: none"> Format DECLARATIVES. <i>section-name</i> SECTION [<i>priority-number</i>]. USE [<i>paragraph-name</i>. [<i>sentence</i>]] END DECLARATIVES. 	V	PS			N
ACCEPT	<ul style="list-style-type: none"> Format 1: data transfer ACCEPT <i>id</i> [FROM <<i>mnemonic-name</i> <i>environment-name</i>>] [<[ON] EXCEPTION <i>imperative-statement-1</i> NOT [ON] EXCEPTION <i>imperative-statement-2</i>>] [END-ACCEPT] 	V	S		<i>id</i>	Y
	<ul style="list-style-type: none"> Format 2: system information transfer ACCEPT <i>id</i> FROM <DATE [YYYYMMDD] DAY [YYYYDDD] DAY-OF-WEEK TIME> 	V	S		<i>id</i>	N
ADD	<ul style="list-style-type: none"> Format 1 ADD <<i>id-1</i> <i>literal</i>> TO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with GIVING ADD <<i>id-1</i> <i>literal-1</i>> TO <<i>id-2</i> <i>literal-1</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S
	<ul style="list-style-type: none"> Format 3: with CORRESPONDING ADD <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-ADD] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
ALTER	<ul style="list-style-type: none"> Format ALTER <i>procedure-name-1</i> TO [PROCEED TO] <i>procedure-name-2</i> 	V	S			N
		VO	N			N
CALL	<ul style="list-style-type: none"> Format 1: with ON OVERFLOW CALL <<i>id-1</i> <i>literal-1</i>> [USING [BY] <REFERENCE [<<i>id-2</i> <i>file-name-1</i>>] VALUE [<i>id-3</i>] CONTENT [<<i>id-4</i> <i>literal-2</i>>]] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [END-CALL] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i>	<i>id-2</i>	S
	<ul style="list-style-type: none"> Format 2: with ON EXCEPTION CALL <<i>id-1</i> <i>literal-1</i>> [USING [BY] <REFERENCE [<<i>id-2</i> <i>file-name-1</i>>] VALUE [<i>id-3</i>] CONTENT [<<i>id-4</i> <i>literal-2</i>>]] [[ON] EXCEPTION <i>imperative-stmt-1</i>] [NOT [ON] EXCEPTION <i>imperative-stmt-2</i>] [END-CALL] 	VO	S			S
	<ul style="list-style-type: none"> Format 3: with CALL UPON SYSTEM CALL UPON SYSTEM USING <<i>id-1</i> <i>literal-1</i>> [<i>id-2</i>] [STATUS <i>id-3</i>] [[ON] EXCEPTION <i>imperative-stmt-1</i>] [NOT [ON] EXCEP-TION <i>imperative-stmt-2</i>] [END-CALL] 	VO	N			N
CANCEL	<ul style="list-style-type: none"> Format CANCEL <<i>id-1</i> <i>literal-1</i>> 	V	S			N
CLOSE	<ul style="list-style-type: none"> Format 1: sequential CLOSE <i>file-name-1</i> [<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND] [WITH] <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files CLOSE <i>file-name-1</i> [[WITH] LOCK] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N
	<ul style="list-style-type: none"> Format 3: line-sequential files CLOSE <i>filename-1</i> [<REEL UNIT> [[FOR] REMOVAL WITH NO REWIND] [WITH] <NO REWIND LOCK>] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
		VO	N			N

1-278 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
COMPUTE	<ul style="list-style-type: none"> Format COMPUTE <i>id-1</i> [ROUNDED] [EQUAL] <i>arithmetic-expr</i> [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-COMPUTE] 	V	S	<i>arithmetic-expr</i>	<i>id-1</i>	S
CONTINUE	<ul style="list-style-type: none"> Format CONTINUE 	V	S			N
DELETE	<ul style="list-style-type: none"> Format DELETE <i>file-name-1</i> [RECORD] [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-DELETE] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
DISPLAY	<ul style="list-style-type: none"> Format DISPLAY <<i>id-1</i> <i>integer-1</i>> [UPON <i>mnemonic-name-1</i>] [[ON] EXCEPTION <i>imperative-stmt-1</i>] [NOT [ON] EXCEPTION <i>imperative-stmt-2</i>] [END-DISPLAY] 	V	S	<i>id-1</i>		S
DIVIDE	<ul style="list-style-type: none"> Format 1: INTO DIVIDE <<i>id-1</i> <i>literal-1</i>> INTO <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-DIVIDE] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-2</i>	S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
	<ul style="list-style-type: none"> Format 2: INTO with GIVING DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3: BY DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 4: INTO with REMAINDER DIVIDE <id-1 literal-1> INTO <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
	<ul style="list-style-type: none"> Format 5: BY with REMAINDER DIVIDE <id-1 literal-1> BY <id-2 literal-2> GIVING id-3 [ROUNDED] REMAINDER id-4 [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-DIVIDE] 	V	S	id-1 id-2	id-3 id-4	S
ENTRY	<ul style="list-style-type: none"> Format 1 ENTRY literal-1 [USING [[BY] REFERENCE] id-1]] [BY] VALUE] id-1] 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 2 ENTRY literal-1 [USING [[BY] VALUE] id-1] 	VO	N			N

1-280 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
EVALUATE	<ul style="list-style-type: none"> Format EVALUATE <id-1 literal-1 expr-1 TRUE FALSE> [ALSO <id-2 literal-2 expr-2 TRUE FALSE>] WHEN phrase-1 [ALSO phrase-2] imperative-stmt-1 [WHEN OTHER imperative-stmt-2] [END-EVALUATE] Phrases phrase-1 and phrase-2 should be represented in the following form: <ANY condition-1 subcondition-1 TRUE FALSE [NOT] <id-3 literal-1 arithmetic-expr-1> [<THROUGH THRU> <id-4 literal-2 arithmetic-expr-2>]> 	V	S	id-1 expr-1 id-2 expr-2 condition-1 id-3 id-4 arithmetic-expr1 arithmetic-expr2		N
EXIT	<ul style="list-style-type: none"> Format EXIT [[TO] TEST [OF]] PERFORM 	V	S			N
EXIT METHOD	<ul style="list-style-type: none"> Format EXIT METHOD. 	N	N			N
EXIT PROGRAM	<ul style="list-style-type: none"> Format EXIT PROGRAM. 	V	S			N
GOBACK	<ul style="list-style-type: none"> Format GOBACK 	V	S			N
GO TO	<ul style="list-style-type: none"> Format 1: unconditional GO [TO] procedure-name-1 	V	S			N
	<ul style="list-style-type: none"> Format 2: conditional GO [TO] procedure-name-1 DEPENDING [ON] id-1 	V	S	id-1		N
	<ul style="list-style-type: none"> Format 3: altered paragraph-name. GO [TO]. 	V	S			N
	<ul style="list-style-type: none"> Format 4: MORE-LABELS GO [TO] MORE-LABELS 	N	N			N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
IF	<ul style="list-style-type: none"> Format IF <i>condition-1</i> THEN <stmt-1 NEXT SENTENCE> [ELSE <stmt-2 NEXT SENTENCE>] [END-IF] 	V	S	<i>condition-1</i>		N
INITIALIZE	<ul style="list-style-type: none"> Format INITIALIZE <i>id-1</i> [REPLACING <ALPHABETIC ALPHANUMERIC NUMERIC ALPHANUMERIC-EDITED NUMERIC-EDITED DBCS EGCS> [DATA] BY <<i>id-2</i> <i>literal-1</i>>] 	V	S	<i>id-2</i>	<i>id-1</i>	N
INSPECT	<ul style="list-style-type: none"> Format 1 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-2</i>		N
	<ul style="list-style-type: none"> Format 2 INSPECT <i>id-1</i> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> 	V	S	<i>id-1</i> <i>id-5</i> <i>id-4</i> <i>id-3</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 3 INSPECT <i>id-1</i> TALLYING <i>id-2</i> FOR <CHARACTERS [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>] <ALL LEADING> <<i>id-3</i> <i>literal-1</i>> [<BEFORE AFTER> [INITIAL] <<i>id-4</i> <i>literal-2</i>>]> REPLACING <CHARACTERS BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>] <ALL LEADING FIRST> <<i>id-3</i> <i>literal-1</i>> BY <<i>id-5</i> <i>literal-3</i>> [<BEFORE AFTER> [INITIAL] <<i>id-6</i> <i>literal-4</i>>]> 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i> <i>id-5</i> <i>id-6</i>	<i>id-1</i>	N
	<ul style="list-style-type: none"> Format 4 INSPECT <i>id-1</i> CONVERTING <<i>id-6</i> <i>literal-4</i>> TO <<i>id-7</i> <i>literal-5</i>> [BEFORE AFTER] [INITIAL] <<i>id-4</i> <i>literal-2</i>> 	V	S	<i>id-1</i> <i>id-6</i> <i>id-7</i> <i>id-4</i>	<i>id-1</i>	N
INVOKE	<ul style="list-style-type: none"> Format INVOKE <<i>id-1</i> <i>class-name-1</i> SELF [<i>class-name-2</i> OF] SUPER> <<i>literal-1</i> <i>id-2</i>> [USING <[[BY] REFERENCE] <[ADDRESS OF] <i>id-3</i> OMITTED> [[BY] CONTENT] <[ADDRESS OF LENGTH OF] <i>id-4</i> <i>literal-2</i> OMITTED> [[BY] VALUE] <[ADDRESS OF LENGTH OF] <i>id-5</i> <i>literal-3</i>>>] [RETURNING <i>id-6</i>] [[ON] EXCEPTION <i>imperative-stmt-1</i>] [NOT [ON] EXCEPTION <i>imperative-stmt-2</i>] [END-INVOKE] 	N	N			N

1-282 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
MERGE	<ul style="list-style-type: none"> Format MERGE <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [<KEY KEY-YY>] <i>data-name-1</i> [[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>] 	V	S	<i>file-name-1</i> <i>data-name-1</i>	<i>file-name-4</i>	N
	<ul style="list-style-type: none"> Format USING <i>file-name-2</i> <i>file-name-3</i> <OUTPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] GIVING <i>file-name-4</i>> 	VO	N	<i>file-name-2</i> <i>file-name-3</i>		N
	<ul style="list-style-type: none"> Format MOVE <<i>id-1</i> <i>literal-1</i>> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	S
MOVE	<ul style="list-style-type: none"> Format 2 MOVE <CORRESPONDING CORR> <i>id-1</i> TO <i>id-2</i> 	V	S	<i>id-1</i>	<i>id-2</i>	N
	<ul style="list-style-type: none"> Format 1 MULTIPLY <<i>id-1</i> <i>literal-1</i>> BY <i>id-2</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-MULTIPLY] 	V	S	<i>id-1</i>	<i>id-2</i>	S
MULTIPLY	<ul style="list-style-type: none"> Format 2: with GIVING MULTIPLY <<i>id-1</i> <i>literal-1</i>> BY <<i>id-2</i> <i>literal-2</i>> GIVING <i>id-3</i> [ROUNDED] [[ON] SIZE ERROR <i>imperative-stmt-1</i>] [NOT [ON] SIZE ERROR <i>imperative-stmt-2</i>] [END-MULTIPLY] 	V	S	<i>id-1</i> <i>id-2</i>	<i>id-3</i>	S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
OPEN	<ul style="list-style-type: none"> Format 1: sequential files OPEN <INPUT <i>file-name-1</i> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N
	[REVERSED WITH NO REWIND]	V	N			N
	OUTPUT <i>file-name-2</i>	V	S			N
	[WITH NO REWIND]	V	N			N
	I-O <i>file-name-3</i> EXTEND <i>file-name-4</i> >	V	S			N
	<ul style="list-style-type: none"> Format 2: indexed and relative files OPEN <INPUT <i>file-name-1</i> OUTPUT <i>file-name-2</i> I-O <i>file-name-3</i> EXTEND <i>file-name-4</i>> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N
<ul style="list-style-type: none"> Format 3: line-sequential files OPEN <INPUT <i>file-name-1</i> OUTPUT <i>file-name-2</i> EXTEND <i>file-name-4</i>> 	V	S	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	<i>file-name-1</i> <i>file-name-2</i> <i>file-name-3</i> <i>file-name-4</i>	N	
PERFORM	<ul style="list-style-type: none"> Format 1: basic PERFORM <<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <i>imperative-stmt-1</i> END-PERFORM> 	V	S			N
	<ul style="list-style-type: none"> Format 2: with TIMES phrase PERFORM <<i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <<i>id-1</i> <i>integer-1</i>> TIMES <i>imperative-stmt-1</i> END-PERFORM> 	V	S	<i>id-1</i>		N
	<ul style="list-style-type: none"> Format 3: with UNTIL phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] UNTIL <i>condition-1</i> <i>imperative-stmt-1</i> END-PERFORM 	V	S	<i>condition-1</i>		N
	<ul style="list-style-type: none"> Format 4: with UNTIL phrase, without END-PERFORM PERFORM <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] [[WITH] TEST <BEFORE AFTER>] UNTIL <i>condition-1</i> 	S	S	<i>condition-1</i>		N

1-284 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
PERFORM	<ul style="list-style-type: none"> Format 5: with VARYING phrase and END-PERFORM PERFORM [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 imperative-stmt-1 END-PERFORM 	V	S	id-3 id-4 condition-1	id-2	N
	<ul style="list-style-type: none"> Format 6: with VARYING phrase, without END-PERFORM PERFORM procedure-name-1 [<THROUGH THRU> procedure-name-2] [[WITH] TEST <BEFORE AFTER>] VARYING <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-1 [AFTER <id-5 index-name-3> FROM <id-6 index-name-4 literal-3> BY <id-7 literal-4> UNTIL condition-2] <id-2 index-name-1> FROM <id-3 index-name-2 literal-1> BY <id-4 literal-2> UNTIL condition-2 	N	N			N
READ	<ul style="list-style-type: none"> Format 1: sequential retrieval READ file-name-1 [[WITH] NO [LOCK]] [NEXT] [RECORD] 	V	S	file-name-1 id-1	file-name-1 id-1	N
	[NEXT] [RECORD]	VO	N			N
	[INTO id-1] [[AT] END imperative-stmt-1] [NOT [AT] END imperative-stmt-2] [END-READ]	V	S			S
	<ul style="list-style-type: none"> Format 2: random retrieval READ file-name-1 [RECORD] [INTO id-1] [KEY [IS] data-name-1] [INVALID [KEY] imperative-stmt-3] [NOT INVALID [KEY] imperative-stmt-4] [END-READ] 	V	S	file-name-1 data-name-1	file-name-1 id-1	N
RELEASE	<ul style="list-style-type: none"> Format RELEASE record-name-1 [FROM id-1] 	V	S	record-name-1 id-1	record-name-1	N
RETURN	<ul style="list-style-type: none"> Format RETURN file-name-1 [RECORD] [INTO id-1] [[AT] END imperative-stmt-1] [NOT [AT] END imperative-stmt-2] [END-RETURN] 	V	S	file-name-1	id-1	N
RE-WRITE	<ul style="list-style-type: none"> Format REWRITE file-name-1 [FROM id-1] [INVALID [KEY] imperative-stmt-1] [NOT INVALID [KEY] imperative-stmt-2] [END-REWRITE] 	V	S	file-name-1 id-1	file-name-1	N

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
SEARCH	<ul style="list-style-type: none"> Format 1: serial search SEARCH <i>id-1</i> [VARYING <<i>id-2</i> <i>index-name-1</i>>] [[AT] END <i>imperative-stmt-1</i>] WHEN <i>condition-1</i> <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>condition-1</i>	<i>id-2</i>	N
	<ul style="list-style-type: none"> Format 2: binary search SEARCH ALL <i>id-1</i> [[AT] END <i>imperative-stmt-1</i>] WHEN <<i>data-name-1</i> [IS] EQUAL [TO] <<i>id-3</i> <i>literal-1</i> <i>arithmetic-expr-1</i>> <i>condition-name-1</i>> [AND <<i>data-name-2</i> [IS] EQUAL [TO] <<i>id-4</i> <i>literal-2</i> <i>arithmetic-expr-2</i>> <i>condition-name-2</i>>] <<i>imperative-stmt-2</i> NEXT SENTENCE> [END-SEARCH] 	V	S	<i>id-1</i> <i>data-name-1</i>		N
SET	<ul style="list-style-type: none"> Format 1: basic table handling SET <<i>index-name-1</i> <i>id-1</i>> TO <<i>index-name-2</i> <i>id-2</i> <i>integer-1</i> ZEROS SEROES> 	V	S	<i>index-name-2</i> <i>id-2</i>	<i>index-name-1</i> <i>id-1</i>	N
	<ul style="list-style-type: none"> Format 2: adjusting indexes SET <i>index-name-3</i> <UP BY DOWN BY> <<i>id-3</i> <i>integer-2</i>> 	V	S	<i>index-name-3</i> <i>id-3</i>	<i>index-name-3</i>	N
	<ul style="list-style-type: none"> Format 3: external switches SET <i>mnemonic-name-1</i> TO <ON OFF> 	V	S		<i>mnemonic-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: condition-names SET <i>condition-name-1</i> [<i>condition-name-2</i>] TO <TRUE FLASE> 	V	S		<i>condition-variable</i>	S
	<ul style="list-style-type: none"> Format 5: USAGE IS POINTER data items SET <<i>id-4</i> ADDRESS OF <i>id-5</i>> TO <<i>id-6</i> ADDRESS OF <i>id-7</i> NULL NULLS> 	V	S	<i>id-6</i> <i>id-7</i>	<i>id-4</i> <i>id-5</i>	N
	<ul style="list-style-type: none"> Format 6: USAGE IS PROCEDURE-POINTER data items SET <i>procedure-pointer-data-item-1</i> TO <<i>procedure-pointer-data-item-2</i> ENTRY <<i>id-8</i> <i>literal-1</i>> NULL NULLS <i>pointer-data-item-3</i>> 	VO	N			N
	<ul style="list-style-type: none"> Format 7: USAGE OBJECT REFERENCE data items SET <i>object-reference-id-1</i> TO <<i>object-reference-id-2</i> NULL SELF> 	N	N			N

1-286 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
SORT	<ul style="list-style-type: none"> Format SORT <i>file-name-1</i> [ON] <ASCENDING DESCENDING> [<KEY KEY-YY>] <i>data-name-1</i> [[WITH] DUPLICATES [IN] [ORDER]] 	V	S	<i>file-name-1</i> <i>data-name-1</i> <i>file-name-2</i>	<i>file-name-3</i>	N
	[[COLLATING] SEQUENCE [IS] <i>alphabet-name-1</i>]	VO	N			N
	<USING <i>file-name-2</i> INPUT PROCEDURE [IS] <i>procedure-name-1</i> [<THROUGH THRU> <i>procedure-name-2</i>] <GIVING <i>file-name-3</i> OUTPUT PROCEDURE [IS] <i>procedure-name-3</i> [<THROUGH THRU> <i>procedure-name-4</i>]>	V	S			N
START	<ul style="list-style-type: none"> Format START <i>file-name-1</i> [WITH NO LOCK] [KEY [IS] <EQUAL [TO] = LESS [THAN] < GREATER [THAN] > NOT LESS [THAN] NOT < NOT GREATER [THAN] NOT > LESS [THAN] OR EQUAL [TO] <= GREATER [THAN] OR EQUAL [TO] >= >] <i>data-name-1</i> [INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>] [END-START] 	V	S	<i>file-name-1</i>	<i>file-name-1</i>	N
STOP	<ul style="list-style-type: none"> Format STOP <RUN <i>literal</i>> 	V	S			N
STRING	<ul style="list-style-type: none"> Format STRING <<i>id-1</i> <i>literal-1</i>> [DELIMITED [BY]] <<i>id-2</i> <i>literal-2</i> SIZE> INTO <i>id-3</i> [[WITH] POINTER <i>id-4</i>] [[ON] OVERFLOW <i>imperative-stmt-1</i>] [NOT [ON] OVERFLOW <i>imperative-stmt-2</i>] [END-STRING] 	V	S	<i>id-1</i> <i>id-2</i> <i>id-3</i> <i>id-4</i>	<i>id-3</i> <i>id-4</i>	S

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Writ-ten' variables	DB
SUBTRACT	<ul style="list-style-type: none"> Format 1 SUBTRACT <id-1 literal-1> FROM id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	S
	<ul style="list-style-type: none"> Format 2: with GIVING SUBTRACT <id-1 literal-1> FROM <id-2 literal-2> GIVING id-3 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-3	S
	<ul style="list-style-type: none"> Format 3 SUBTRACT <CORRESPONDING CORR> id-1 FROM id-2 [ROUNDED] [[ON] SIZE ERROR imperative-stmt-1] [NOT [ON] SIZE ERROR imperative-stmt-2] [END-SUBTRACT] 	V	S	id-1 id-2	id-2	N
UNSTRING	<ul style="list-style-type: none"> Format UNSTRING id-1 [DELIMITED [BY] [ALL] <id-2 literal-1> [OR [ALL] <id-3 literal-2>]] INTO id-4 [DELIMITER [IN] id-5] [COUNT [IN] id-6] [[WITH] POINTER id-7] [TALLYING [IN] id-8] [[ON] OVERFLOW imperative-stmt-1] [NOT [ON] OVERFLOW imperative-stmt-2] [END-UNSTRING] 	V	S	id-1 id-2 id-3 id-4 id-7 id-8	id-4 id-5 id-6 id-7 id-8	S

1-288 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statement	Format	Parser	CE			
			Gen.	'Read' variables	'Written' variables	DB
WRITE	<ul style="list-style-type: none"> Format 1: sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [[INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>]] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 2: sequential files with 'BEFORE/AFTER' WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [<<BEFORE AFTER> [ADVANCING] <<<i>id-1</i> <i>integer-1</i>> [LINE LINES] <i>mnemonic-name-1</i> PAGE>] [[AT] <END-OF-PAGE EOP> <i>imperative-stmt-3</i>] [NOT [AT] <END-OF-PAGE EOP> <i>imperative-stmt-4</i>] [[INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>]] [END-WRITE] 	VO	N			N
	<ul style="list-style-type: none"> Format 3: indexed and relative files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [[INVALID [KEY] <i>imperative-stmt-1</i>] [NOT INVALID [KEY] <i>imperative-stmt-2</i>]] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 4: line-sequential files WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [END-WRITE] 	V	S	<i>record-name-1</i> <i>id-1</i>	<i>record-name-1</i>	N
	<ul style="list-style-type: none"> Format 5: line-sequential files with 'AFTER' WRITE <i>record-name-1</i> [FROM <i>id-1</i>] [AFTER [ADVANCING] <<i>id-2</i> <i>integer-1</i>> [LINE LINES] PAGE>] [END-WRITE] 	VO	N			N

Intrinsic functions

Functions	Parser	CE	
		Gen.	DB
ACOS	V	S	N
ANNUITY	V	S	N
ASIN	V	S	N
ATAN	V	S	N
CHAR	V	S	N
COS	V	S	N
CURRENT-DATE	V	S	N
DATE-OF-INTEGER	V	S	N
DATE-TO-YYYYMMDD	V	S	N
DATEVAL	V	S	N
DAY-OF-INTEGER	V	S	N
DAY-TO-YYYYDDD	V	S	N
FACTORIAL	V	S	N
INTEGER	V	S	N
INTEGER-OF-DATE	V	S	N
INTEGER-OF-DAY	V	S	N
INTEGER-PART	V	S	N
LENGTH	V	S	N
LOG	V	S	N
LOG10	V	S	N
LOWER-CASE	V	S	N

1-290 Supported COBOL Statements
Supported SIEMENS COBOL statements

Functions	Parser	CE	
		Gen.	DB
MAX	V	S	N
MEAN	V	S	N
MEDIAN	V	S	N
MIDRANGE	V	S	N
MIN	V	S	N
MOD	V	S	N
NUMVAL	V	S	N
NUMVAL-C	V	S	N
ORD	V	S	N
ORD-MAX	V	S	N
ORD-MIN	V	S	N
PRESENT-VALUE	V	S	N
RANDOM	V	S	N
RANGE	V	S	N
REM	V	S	N
REVERSE	V	S	N
SIN	V	S	N
SQRT	V	S	N
STANDARD-DEVIATION	V	S	N
SUM	V	S	N
TAN	V	S	N
UNDATE	V	S	N

Functions	Parser	CE	
		Gen.	DB
UPPER-CASE	V	S	N
VARIANCE	V	S	N
WHEN-COMPILED	V	S	N
YEAR-TO-YYYY	V	S	N
YEARWINDOW	V	S	N

Compile-directing statements and directives

Statements and directives	Format	Parser	CE
BASIS	<ul style="list-style-type: none"> Format [sequence-number] BASIS <basis-name literal-1> 	VO	N
CBL (PROCESS)	<ul style="list-style-type: none"> Format <CBL PROCESS> [options-list] 	VO	N
CALLINTER-FACE directive	<ul style="list-style-type: none"> Format < >>CALLINTERFACE >>CALLINT > [SYSTEM OPTLINK FAR16 PASCAL16 CDECL] [DESC DESCRIPTOR NODESC NODESCRIPTOR] 	VO	N
*CONTROL (*CBL)	<ul style="list-style-type: none"> Format <*CONTROL *CBL> <SOURCE NOSOURCE LIST NOLIST MAP NOMAP> 	VO	N
COPY	<ul style="list-style-type: none"> Format COPY <text-name-1 literal-1> [<OF IN> <library-name literal-2>] [SUPPRESS] [REPLACING operand-1 BY operand-2] 	V	S
DELETE	<ul style="list-style-type: none"> Format [sequence-number] DELETE sequence-number-field 	VO	N
EJECT	<ul style="list-style-type: none"> Format EJECT [.] 	VO	N

1-292 Supported COBOL Statements
Supported SIEMENS COBOL statements

Statements and directives	Format	Parser	CE
ENTER	<ul style="list-style-type: none"> Format ENTER <i>language-name-1</i> [<i>routine-name-1</i>]. 	VO	N
INSERT	<ul style="list-style-type: none"> Format [<i>sequence-number</i>] INSERT <i>sequence-number-field</i> 	VO	N
READY or RESET TRACE	<ul style="list-style-type: none"> Format <READY RESET> TRACE. 	VO	N
REPLACE	<ul style="list-style-type: none"> Format 1 REPLACE ==<i>pseudo-text-1</i>== BY ==<i>pseudo-text-2</i>==. 	V	S
	<ul style="list-style-type: none"> Format 2 REPLACE OFF. 	V	S
SERVICE LABEL	<ul style="list-style-type: none"> Format SERVICE LABEL 	VO	N
SERVICE RELOAD	<ul style="list-style-type: none"> Format SERVICE RELOAD <i>id-1</i> 	VO	N
SKIP1/2/3	<ul style="list-style-type: none"> Format <SKIP1 SKIP2 SKIP3>. 	VO	N
TITLE	<ul style="list-style-type: none"> Format TITLE <i>literal</i>. 	VO	N
USE	<ul style="list-style-type: none"> Format 1: EXCEPTION ERROR declarative USE [GLOBAL] AFTER [STANDARD] <EXCEPTION ERROR> PROCEDURE [ON] <<i>file-name-1</i> INPUT OUTPUT I-O EXTEND> 	VO	N
	<ul style="list-style-type: none"> Format 2: LABEL declarative USE [GLOBAL] AFTER [STANDARD] [BEGINNING ENDING] [FILE REEL UNIT] LABEL PROCEDURE [ON] <<i>file-name-1</i> INPUT OUTPUT I-O EXTEND> 	VO	N

Supported Natural Statements



This chapter describes the Natural Programming Language statements supported by ATW for component extraction. For a list of supported legacy versions, refer to the *Release Notes*.

Key to tables

The tables below contain detailed information on ATW component extraction support. The first column of each table presents Natural keywords. Next, if needed, in the 'Format' column, the possible forms of usage for each keyword are explained. These forms are represented in the standard grammar notation, where

- the optional arguments are written in square brackets '[arguments]',
- the arguments which can not be omitted are listed in angle brackets <arguments>,
- the vertical line '|' means 'or' logical connective, and

2-2 Supported Natural Statements
Key to tables

- dots ‘...’ mean that last argument or clause can be repeated many times.

In this way, the notation <arg1 | arg2 | arg3> means that one of the arg1, arg2 or arg3 is required. Other columns provide the following information:

- **Parser** — ATW Parser. The possible values in this column are:
 - (V)erified — parsed and prepared for further processing. Possibly is supported by other ATW tools. Does not initiate any errors or warnings during the verification phase.
 - (V)erification (O)nly — parsed and skipped immediately. Further processing is not possible, i.e. not supported by ATW tools. Verification warning is initiated.
 - (N)ot Verified — if the ATW parser finds such an entity it stops and verification is considered as erroneous.
- **CE** — Component Extraction. Component Extraction (S)upport means that the entity can be correctly sliced. If the entity is (N)ot supported and is represented in the slicing code the obtained program may not work correctly. The ‘Gen.’ column gives information on ATW support for:
 - Computation-Based Component Extraction
 - Dead Code Elimination
 - External Subroutine Component Extraction

The tables below also provide information on ‘Read’ and ‘Write’ variables. This is the knowledge which is used by ATW to simplify the slicing program. If the variable is situated in ‘Read’ column, then ATW ‘knows’ that it is used when corresponding statement is executed. If it is in the ‘Write’ column, it will be modified when operation is performed.

Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Constants

Constant type	Format	Parser	CE
Alphanumeric constants	<ul style="list-style-type: none"> Format 1: with quotes Example: "THIS ISN'T WRONG" 	V	S
	<ul style="list-style-type: none"> Format 2: with apostrophes Example: 'THIS ISN'T WRONG' 	V	S
Numeric constants	<ul style="list-style-type: none"> Format 1: fixed-point Example: -2.71828 	V	S
	<ul style="list-style-type: none"> Format 2: floating-point <+ -> mantissa E <+ -> exponent 	V	S
Hexadecimal constants	<ul style="list-style-type: none"> Format Example: H'A1A2A3' 	V	S
Logical constants	<ul style="list-style-type: none"> Format <TRUE FALSE> 	V	S
Date and time constants	<ul style="list-style-type: none"> Format 1 Example: D'DD.MM.YYYY' 	V	S
	<ul style="list-style-type: none"> Format 2 Example: T'hh:mm:ss' 	V	S
	<ul style="list-style-type: none"> Format 3 E"DD.MM.YYYY hh:mm:ss' 	N	N
Attribute constants	<ul style="list-style-type: none"> Format ([AD-D AD-B AD-I AD-N AD-V AD-U AD-C AD-Y AD-P] [CD-BL CD-GR CD-NE CD-PI CD-RE CD-TU CD-YE]) 	N	N

2-4 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

System variables

Variable	Parser	CE
*APPLIC-ID	V	S
*APPLIC-NAME	V	S
*COM	V	S
*CONTROL	N	N
*CONVID	V	S
*COUNTER	V	S
*CPU-TIME	N	N
*CURS-COL	V	S
*CURS-FIELD	V	S
*CURS-LINE	V	S
*CURSOR	V	S
*DATA	V	S
*DEVICE	V	S
*DIALOG-ID	V	S
*ERROR-LINE	V	S
*ERROR-NR	V	S
*ERROR-TA	V	S
*ETID	V	S
*EVENT	V	S
*GROUP	V	S
*HARDCOPY	V	S
*HARDWARE	N	N

Variable	Parser	CE
*HOSTNAME	N	N
*INIT-ID	V	S
*INIT-PROGRAM	V	S
*INIT-USER	V	S
*ISN	V	S
*LANGUAGE	V	S
*LENGTH	N	N
*LEVEL	V	S
*LIBRARY-ID	V	S
*LINE-COUNT	V	S
*LINESIZE	V	S
*LOG-LS	N	N
*LOG-PS	N	N
*MACHINE-CLASS	N	N
*NATVERS	N	N
*NET-USER	N	N
*NUMBER	V	S
*OCCURRENCE	V	S
*OPSYS	N	N
*OS	N	N
*OSVERS	N	N
*PAGESIZE	V	S
*PAGE-NUMBER	V	S

2-6 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Variable	Parser	CE
*PARM-USER	N	N
*PATCH-LEVEL	N	N
*PF-KEY	V	S
*PF-NAME	V	S
*PID	N	N
*PROGRAM	V	S
*ROWCOUNT	N	N
*SCREEN-IO	N	N
*SERVER-TYPE	N	N
*STARTUP	V	S
*STEPLIB	V	S
*SUBROUTINE	N	N
*THIS-OBJECT	N	N
*TPSYS	N	N
*UI	N	N
*USER	V	S
*USER-NAME	V	S
*WINDOW-LS	N	N
*WINDOW-POS	N	N
*WINDOW-PS	N	N
*WINMGR	N	N
*WINMGRVERS	N	N

Date and time system variables

Variable	Parser	CE
*DATD	V	S
*DAT4D	N	N
*DATE	V	S
*DAT4E	V	S
*DATG	V	S
*DATI	V	S
*DAT4I	N	N
*DATJ	V	S
*DAT4J	N	N
*DATN	V	S
*DATU	V	S
*DAT4U	N	N
*DATV	N	N
*DATVS	N	N
*DATX	V	S

Session parameters

Parameter	Parser	CE
AD	V	S
AL	V	S
BX	V	S

Parameter	Parser	CE
FL	V	S
FS	V	S
GC	V	S

Parameter	Parser	CE
MT	V	S
NC	V	S
NL	V	S

2-8 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

CC	V	S
CD	V	S
CF	V	S
CO	V	S
CV	V	S
DC	V	S
DF	V	S
DFOUT	V	S
DFSTACK	V	S
DFTITLE	V	S
DU	V	S
DY	V	S
EJ	V	S
EM	V	S
ES	V	S
FC	V	S
FCDP	V	S

HC	V	S
HE	V	S
HW	V	S
IA	V	S
IC	V	S
ID	V	S
IM	V	S
IP	V	S
IS	V	S
KD	V	S
LC	V	S
LE	V	S
LS	V	S
LT	V	S
MC	V	S
MP	V	S
MS	V	S

OPF	V	S
PC	V	S
PD	V	S
PM	V	S
PS	V	S
REINP	V	S
SA	V	S
SF	V	S
SG	V	S
SL	V	S
SM	V	S
TC	V	S
TS	V	S
UC	V	S
WH	V	S
ZD	V	S
ZP	V	S

System functions

System function	Parser	CE
AVER	V	S
COUNT	V	S
MAX	V	S
MIN	V	S
NAVER	V	S
NCOUNT	V	S
NMIN	V	S
OLD	V	S
POS	V	S
RET	V	S
SORTKEY	N	N
SUM	V	S
TOTAL	V	S

Mathematical function	Parser	CE
ABS	V	S
ATN	V	S
COS	V	S
EXP	V	S
FRAC	V	S
INT	V	S
LOG	V	S
SGN	V	S
SIN	V	S
SQRT	V	S
TAN	V	S
VAL	V	S

2-10 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Conditional expressions

Conditions	Format	Parser	CE		
			Gen.	Read variables	Written variables
Relation condition	<ul style="list-style-type: none"> Format 1: <i>operand-1</i> <= EQ EQUAL [TO] <> NE NOT = NOT EQ NOT EQUAL [TO] < LT LESS THAN <= LE LESS EQUAL > GT GREATER THAN >= GE GREATER EQUAL> <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>	
	<ul style="list-style-type: none"> Format 2: with SUBSTRING <SUBSTRING (<i>operand-1</i>, <i>op-3</i>, <i>op-4</i>) <i>operand-1</i>> relation condition <SUBSTRING (<i>operand-2</i>, <i>op-5</i>, <i>op-6</i>) <i>operand-2</i>> 	V	S	<i>operand-1</i> <i>operand-2</i> <i>op-3</i> , <i>op-4</i> <i>op-5</i> , <i>op-6</i>	
	<ul style="list-style-type: none"> Format 3: extended <i>op-1</i> <= EQ EQUAL [TO]> <i>op-2</i> <OR <= EQ EQUAL [TO]> <i>op-3</i> THRU <i>op-4</i> [BUT NOT <i>op-5</i> [THRU <i>op-6</i>]]> 	V	S	<i>op-1</i> , <i>op-2</i> <i>op-3</i> , <i>op-4</i> <i>op-5</i> , <i>op-6</i>	
MASK option	<ul style="list-style-type: none"> Format <i>operand-1</i> <= EQ EQUAL [TO] NE NOT EQUAL> MASK (<i>mask-definition</i>) [<i>operand-2</i>] 	V	S	<i>operand-1</i> <i>operand-2</i>	
SCAN option	<ul style="list-style-type: none"> Format <i>operand-1</i> <= EQ EQUAL [TO] NE NOT EQUAL> SCAN <i>operand-2</i> 	V	S	<i>operand-1</i> <i>operand-2</i>	
IS option	<ul style="list-style-type: none"> Format <i>operand-1</i> IS (<i>format</i>) 	V	S	<i>operand-1</i>	
Logical variable evaluation	<ul style="list-style-type: none"> Format <i>operand-1</i> 	V	S	<i>operand-1</i>	
Modified control variables	<ul style="list-style-type: none"> Format <i>operand-1</i> [NOT] MODIFIED 	V	S	<i>operand-1</i>	
SPECIFIED option	<ul style="list-style-type: none"> Format <i>parameter-name</i> [NOT] SPECIFIED 	V	S	<i>parameter-name</i>	
Complex logical expressions	<ul style="list-style-type: none"> Format [NOT] <<i>logical-condition-criterion</i> (<i>logical-expression</i>)> [<OR AND> <i>logical-expression</i>] 	V	S		

Statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
ACCEPT	<ul style="list-style-type: none"> Format ACCEPT [IF] <i>logical-condition</i> 	V	S		
ADD	<ul style="list-style-type: none"> Format ADD [ROUNDED] <i>operand-1</i> ... TO <i>operand-2</i> 	V	S	<i>operand-1</i> ... <i>operand-2</i>	<i>operand-2</i>
ASSIGN	<ul style="list-style-type: none"> Format 1: structured mode <ASSIGN [ROUNDED] <i>op-1</i> = <<i>arithmetic-expression</i> <i>op-2</i>> <i>op-1</i> := <<i>arithmetic-expression</i> <i>op-2</i>>> 	V	S	<i>op-2</i>	<i>op-1</i>
	<ul style="list-style-type: none"> Format 2: reporting mode [ASSIGN [ROUNDED]] <i>op-1</i> = <<i>arithmetic-expression</i> <i>op-2</i>> 	V	S	<i>op-2</i>	<i>op-1</i>
AT BREAK	<ul style="list-style-type: none"> Format 1: structured mode [AT] BREAK [(<i>r</i>)] [OF] <i>op-1</i> [/n] <i>statement</i>... END-BREAK 	V	S	<i>op-1</i>	
	<ul style="list-style-type: none"> Format 2: reporting mode [AT] BREAK [(<i>r</i>)] [OF] <i>op-1</i> [/n] <<i>statement</i> DO <i>statement</i>... DOEND> 	V	S	<i>op-1</i>	
AT END OF DATA	<ul style="list-style-type: none"> Format 1: structured mode [AT] END [OF] DATA [(<i>r</i>)] <i>statement</i>... END-ENDDATA 	V	S		
	<ul style="list-style-type: none"> Format 2: reporting mode [AT] END [OF] DATA [(<i>r</i>)] <<i>statement</i> DO <i>statement</i>... DOEND> 	V	S		
AT END OF PAGE	<ul style="list-style-type: none"> Format 1: structured mode [AT] END [OF] PAGE [(<i>rep</i>)] <i>statement</i>... END-ENDPAGE 	V	S		
	<ul style="list-style-type: none"> Format 2: reporting mode [AT] END [OF] PAGE [(<i>rep</i>)] <<i>statement</i> DO <i>statement</i>... DOEND> 	V	S		

2-12 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
AT START OF DATA	<ul style="list-style-type: none"> Format 1: structured mode [AT] START [OF] DATA [(r)] <i>statement...</i> END-START 	V	S		
	<ul style="list-style-type: none"> Format 2: reporting mode [AT] START [OF] DATA [(r)] <<i>statement</i> DO <i>statement...</i> DOEND> 	V	S		
AT TOP OF PAGE	<ul style="list-style-type: none"> Format 1: structured mode [AT] TOP [OF] PAGE [(rep)] <i>statement...</i> END-TOPPAGE 	V	S		
	<ul style="list-style-type: none"> Format 2: reporting mode [AT] TOP [OF] PAGE [(rep)] <<i>statement</i> DO <i>statement...</i> DOEND> 	V	S		
BACKOUT TRANSACTION	<ul style="list-style-type: none"> Format BACKOUT [TRANSACTION] 	V	S		
BEFORE BREAK PROCESSING	<ul style="list-style-type: none"> Format 1: structured mode BEFORE [BREAK] [PROCESSING] <i>statement...</i> END-BEFORE 	V	S		
	<ul style="list-style-type: none"> Format 2: reporting mode BEFORE [BREAK] [PROCESSING] <<i>statement</i> DO <i>statement...</i> DOEND> 	V	S		
CALL	<ul style="list-style-type: none"> Format CALL <i>op-1</i> [USING] [<i>op-2</i>]... 	V	S	<i>op-1</i> <i>op-2</i> ...	<i>op-2</i> ...
CALL FILE	<ul style="list-style-type: none"> Format 1: structured mode CALL FILE '<i>program-name</i>' <i>op-1 op-2 statement...</i>, END-FILE 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-1</i> <i>op-2</i>
	<ul style="list-style-type: none"> Format 2: reporting mode CALL FILE '<i>program-name</i>' <i>op-1 op-2 statement...</i>, [LOOP] 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-1</i> <i>op-2</i>

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
CALL LOOP	<ul style="list-style-type: none"> Format 1: structured mode CALL LOOP <i>op-1</i> [<i>op-2</i>] <i>statement...</i>, END-LOOP 	V	S	<i>op-1</i> <i>op-2</i> ...	<i>op-2</i> ...
	<ul style="list-style-type: none"> Format 2: reporting mode CALL LOOP <i>op-1</i> [<i>op-2</i>] <i>statement...</i>, [LOOP] 	V	S	<i>op-1</i> <i>op-2</i> ...	<i>op-2</i> ...
CALLNAT	<ul style="list-style-type: none"> Format CALLNAT <i>op-1</i> [[USING] <<i>op-2</i> [(AD-<M O A>)] nX>] 	V	S	<i>op-2</i>	<i>op-2</i>
CLOSE CONVERSATION	<ul style="list-style-type: none"> Format CLOSE CONVERSATION <<i>op-1</i> *CONVID ALL> 	V	S	<i>op-1</i>	
CLOSE DIALOG	<ul style="list-style-type: none"> Format CLOSE DIALOG [USING] [DIALOG-ID] <<i>op-1</i> *DIALOG-ID> 	V	S	<i>op-1</i>	
CLOSE PRINTER	<ul style="list-style-type: none"> Format CLOSE PRINTER <(logical-printer-name) (printer-number)> 	V	S		
CLOSE WORK FILE	<ul style="list-style-type: none"> Format CLOSE WORK [FILE] <i>work-file-number</i> 	V	S		
COMPOSE	<ul style="list-style-type: none"> Format COMPOSE [RESETTNG [DATAAREA TEXTAREA MACROAREA ALL]] [MOVING [<i>op-1</i> ...] [TO DATAAREA] [LAST] [STATUS [TO] <i>op-2</i> [<i>op-3</i> [<i>op-4</i> [<i>op-5</i>]]]]] MOVING <<i>op-1</i> ... [TO DATAAREA] LAST> [OUTPUT] TO VARIABLES <i>op-6</i> ... [STATUS [TO] <i>op-2</i> [<i>op-3</i> [<i>op-4</i> [<i>op-5</i>]]]]] MOVING [OUTPUT] TO VARIABLES <i>op-6</i> ... [STATUS [TO] <i>op-2</i> [<i>op-3</i> [<i>op-4</i> [<i>op-5</i>]]]]] [ASSIGNING [TEXTVARIABLE] <i>op-7</i> = <i>op-8</i> ...] [FORMATTING [OUTPUT <(rep) SUPPRESSED CALLING <i>op-9</i> TO VARIABLES [CONTROL <i>op-10</i> <i>op-11</i>] <i>op-12</i> ... DOCUMENT-option>] [INPUT <DATAAREA [FROM <EXIT <i>op-14</i> CABINEIS <i>op-14</i> [PASSW=<i>op-15</i>]> ...] <i>op-13</i> FROM <EXIT <i>op-14</i> CABINETIS <i>op-14</i> [PASSW=<i>op-15</i>]> ...>] [STATUS <i>op-16</i> [<i>op-17</i> [<i>op-18</i> [<i>op-19</i>]]]] [PROFILE <i>op-16</i>] [MESSAGES <[LISTED] [ON] (rep) SUPPRESSED>] [ERRORS <[LISTED] [ON] (rep) INTERCEPTED>] [ENDING <[AT] [PAGE] <i>op-20</i> AFTER <i>op-20</i> [PAGES]>] [STARTING [FROM] [PAGE] <i>op-21</i>] [EXTRACTING [TEXTVARIABLE] <i>op-22</i> = <i>op-23</i> ...]] 	N	N		

2-14 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
COMPRESS	<ul style="list-style-type: none"> Format COMPRESS [NUMERIC] [FULL] <op-1 [(parameter)]> SUBSTRING (op-1, op-3, op-4) [(parameter)]> INTO <op-2 SUBSTRING (op-2, op-5, op-6)> [LEAVING SPACE LEAVING NO [SPACE] WITH [ALL] DELIMITERS [op-7]] 	V	S	op-1, op-3, op-4, op-5, op-6, op-7	op-2
COMPUTE	<ul style="list-style-type: none"> Format 1: structured mode <COMPUTE [ROUNDED] op-1 = <arithmetic-expression op-2> op-1 := <arithmetic-expression op-2>> 	V	N		
	<ul style="list-style-type: none"> Format 2: reporting mode [COMPUTE [ROUNDED]] op-1 = <arithmetic-expression op-2> 	V	N		
CREATE OBJECT	<ul style="list-style-type: none"> Format CREATE OBJECT op-1 OF [CLASS] op-2 [ON [NODE] op-3] [GIVING [op-4]] 	V	S	op-1, op-2, op-3, op-4	
DECIDE FOR	<ul style="list-style-type: none"> Format DECIDE FOR <FIRST EVERY> CONDITION WHEN logical-condition-stmt ... [WHEN ANY stmt ...] [WHEN ALL stmt ...] WHEN NONE stmt ... END-DECIDE 	V	S		
DECIDE ON	<ul style="list-style-type: none"> Format DECIDE ON <FIRST EVERY> [VALUES] [OF] op-1 VALUES <op-2 op-2 ... [op-2]... op-3 : op-4> stmt... [ANY [VALUES] stmt ...] [ALL [VALUES] stmt ...] NONE [VALUES] stmt ... END-DECIDE 	V	S	op-1, op-2, op-3, op-4	
DEFINE CLASS	<ul style="list-style-type: none"> General format (see all definitions below) DEFINE CLASS classname [[WITH] ACTIVATION [POLICY] <ES IM EM>] [OBJECT <USING <local-data-area parameter-data-area> data-definition ...>] ... [LOCAL <USING <local-data-area parameter-data-area> data-definition ...>] ... [ID class-GUID] [INTERFACE USING copycode INTERFACE stmt] ... [PROPERTY stmt] ... [METHOD stmt] ... END-CLASS 	V	S		
DEFINE DATA	<ul style="list-style-type: none"> General format (see all definitions below) DEFINE DATA classname [GLOBAL USING global-data-area [WITH block [.block] ...]] [PARAMETER <USING parameter-data-area parameter-data-definition>] ... [OBJECT <USING <local-data-area parameter-data-area> data-definition ...>] ... [LOCAL <USING <local-data-area parameter-data-area> data-definition ...>] ... [INDEPENDENT AIV-data-definition ...] [CONTEXT <USING <local-data-area parameter-data-area> context-data-definition ...>] 	V	S		
Definition types	<ul style="list-style-type: none"> data-definition level <group-name [(array-definition)] view-definition redefinition variable-definition handle-definition> 	V	S		

Statement	Format	Parser	CE	
			Gen.	'Read' variables
<ul style="list-style-type: none"> • parameter-data-definition <i>level <group-name [(array-definition)] redefinition variable-name <(format-length) (format-length / array-definition) (<A B>) DYNAMIC> [BY VALUE [RESULT] [OPTIONAL] handle-definition></i> • handle-definition <i>handle-name <HANDLE OF <dialog-element-type OBJECT> [<CONSTANT INIT> init-definition] (array-definition) HANDLE OF <dialog-element-type OBJECT> [<CONSTANT INIT> array-init-definition]></i> • view-definition <i>view-name VIEW [OF] ddm-name [level <ddm-field [(format-length) [emhdpm] (<A B>) DYNAMIC] redefinition>]</i> • redefinition <i>REDEFINE field-name level <rfield (format-length) FILLER nX> ...</i> • variable-definition <i>variable-name <(format-length) [<CONSTANT INIT> init-definition] [emhdpm] (format-length / array-definition) [<CONSTANT INIT> array-init-definition] [emhdpm] (<A B>) DYNAMIC></i> • init-definition <i>< <constant> <system-variable> FULL LENGTH <character-s> LENGTH n <character-s> ></i> 	V	S		
	N	N		
	V	S		
	V	S		
	V	S		
	V	S		
<ul style="list-style-type: none"> • array-definition <i>index [:index] , ...</i> • array-init-definition <i>[ALL (<index [:index] V> , ...)] <<FULL LENGTH LENGTH n> <character-s, ...> <constant, ...> <system-variable, ...>></i> • emhdpm <i>[(EM = value) [HD = 'value'] [PM=value]]</i> • AIV-data-definition <i>level <AIV-definition REDEFINE field-name></i> 	V	N		
	V	N		
	V	N		
	V	N		

2-16 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
	<ul style="list-style-type: none"> • AIV-definition <i>variable-name</i> <(format-length) [INIT <i>init-definition</i>] (format-length / array-definition) [INIT <i>array-init-definition</i>]> [emhdpm] 	V	N		
	<ul style="list-style-type: none"> • context-data-definition <i>level</i> <<i>variable-definition</i> <i>redefinition</i> <i>handle-definition</i>> 	V	N	<i>op-1</i> <i>op-2</i>	
DEFINE PRINTER	<ul style="list-style-type: none"> • Format DEFINE PRINTER ([<i>logical-printer-name</i> =] <i>n</i>) [OUTPUT <i>op-1</i>] [PROFILE <i>op-2</i> FORMS <i>op-2</i> NAME <i>op-2</i> DISP <i>op-2</i> CLASS <i>op-2</i> COPIES <i>op-2</i> PRTY <i>op-2</i>] 	V	S		
DEFINE SUB-ROUTINE	<ul style="list-style-type: none"> • Format DEFINE [SUBROUTINE] <i>subroutine-name statement ...</i> <END-SUBROUTINE RETURN> 	V	S		
DEFINE WINDOW	<ul style="list-style-type: none"> • Format DEFINE WINDOW <i>window-name</i> [SIZE <AUTO QUARTER <i>op-1</i> * <i>op-2</i>>] [BASE <CURSOR <TOP BOTTOM> <LEFT RIGHT> <i>op-3</i> / <i>op-4</i>>] [REVERSED [(CD = <i>background-color</i>)]] [TITLE <i>op-5</i>] [CONTROL <WINDOW SCREEN>] [FRAMED <OFF ON] [(CD = <i>frame-color</i>)] [POSITION <SYMBOL [TOP BOTTOM] [AUTO] [SHORT] [LEFT RIGHT] TEXT [MORE] [LEFT RIGHT] OFF>] 	V	S	<i>op-1</i> <i>op-2</i> <i>op-3</i> <i>op-4</i> <i>op-5</i>	
DEFINE WORK FILE	<ul style="list-style-type: none"> • Format DEFINE WORK FILE <i>n op-1</i> [TYPE <i>op-2</i>] 	V	S	<i>op-1</i> <i>op-2</i>	
DELETE	<ul style="list-style-type: none"> • Format DELETE [RECORD] [IN] [STATEMENT] [(<i>r</i>)] 	V	S		
DISPLAY	<ul style="list-style-type: none"> • Format DISLPAY [(<i>rep</i>)] [NOTITLE] [NOHDR] [[AND] [GIVE] [SYSTEM] FUNCTIONS] [(<i>stmt-parameters</i>)] [<i>f ...</i>] [<i>nX</i> <i>nT</i> <i>x / y</i> <i>T*field-name</i> <i>P*field-name</i>] [<i>'text'</i> [(<i>attributes</i>)] '<i>c</i>' (<i>n</i>) [(<i>attributes</i>)]] [VERTICALLY [AS <<i>'text'</i> [(<i>attributes</i>)] [CAPTIONED] CAPTIONED] [<i>f ...</i>] HORIZONTALLY] [<<i>'text'</i> [(<i>attributes</i>)] '<i>c</i>' (<i>n</i>) [(<i>attributes</i>)]> <i>nX</i> <i>nT</i> <i>x / y</i>] [=] <i>op-1</i> [(<i>parameters</i>)] 	V	S	<i>op-1</i>	
DIVIDE	<ul style="list-style-type: none"> • Format 1: basic DIVIDE [ROUNDED] <i>op-1</i> INTO <i>op-2</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-2</i>
	<ul style="list-style-type: none"> • Format 2: with GIVING DIVIDE [ROUNDED] <i>op-1</i> INTO <i>op-2</i> GIVING <i>op-3</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-3</i>

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
	<ul style="list-style-type: none"> Format 3: with REMAINDER DIVIDE <i>op-1</i> INTO <i>op-2</i> REMAINDER <i>op-4</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-2</i> <i>op-4</i>
	<ul style="list-style-type: none"> Format 3: with REMAINDER and GIVING DIVIDE <i>op-1</i> INTO <i>op-2</i> GIVING <i>op-3</i> REMAINDER <i>op-4</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-3</i> <i>op-4</i>
DO / DOEND	<ul style="list-style-type: none"> Format DO <i>stmt</i> ... DOEND 	V	S		
EJECT	<ul style="list-style-type: none"> Format 1 EJECT <ON OFF> [(<i>rep</i>)] 	V	S		
	<ul style="list-style-type: none"> Format 2 EJECT [(<i>rep</i>)] [(IF WHEN] LESS [THAN] <i>op-1</i> [LINES] [LEFT]] 	V	S	<i>op-1</i>	
END	<ul style="list-style-type: none"> Format <END .> 	V	S		
END TRANSACTION	<ul style="list-style-type: none"> Format END [OF] TRANSACTION [<i>op-1</i> ...] 	V	S	<i>op-1</i>	
ESCAPE	<ul style="list-style-type: none"> Format ESCAPE <TOP BOTTOM [(<i>r</i>)] [IMMEDIATE] ROUTINE [IMMEDIATE]> 	V	S		
EXAMINE	<ul style="list-style-type: none"> Format EXAMINE [FULL [VALUE [OF]]] <<i>op-1</i> SUBSTRING (<i>op-1</i>, <i>op-2</i>, <i>op-3</i>)> [FOR] [FULL [VALUE [OF]]] [PATTERN] <i>op-4</i> [ABSOLUTE WITH [DELIMITERS] WITH [DELIMITERS] <i>op-5</i>] [(AND] <DELETE [FIRST] REPLACE [FIRST] [WITH] [FULL [VALUE [OF]]] <i>op-6</i>> [(GIVING] [NUMBER POSITION LENGTH INDEX] [IN] <i>op-7</i>] 	V	S	<i>op-1</i> , <i>op-2</i> , <i>op-3</i> , <i>op-4</i> , <i>op-5</i> , <i>op-6</i>	<i>op-7</i> ; <i>op-1</i> for DELETE and REPLACE
EXAMINE TRANSLATE	<ul style="list-style-type: none"> Format EXAMINE <<i>op-1</i> SUBSTRING (<i>op-1</i>, <i>op-2</i>, <i>op-3</i>)> [AND] TRANSLATE <INTO <UPPER LOWER> CASE USING [(INVERTED] <i>op-4</i>> 	V	S	<i>op-1</i> <i>op-2</i> <i>op-3</i> <i>op-4</i>	<i>op-1</i>

2-18 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
EXPAND	<ul style="list-style-type: none"> Format EXPAND [SIZE OF] DYNAMIC VARIABLE <i>op-1</i> TO <i>op-2</i> 	N	N		
FETCH	<ul style="list-style-type: none"> Format FETCH <REPEAT RETURN> <i>op-1</i> [<i>op-2</i> [(<i>parameter</i>)]] ... 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-2</i>
FIND	<ul style="list-style-type: none"> Format FIND [ALL (<i>op-1</i>) FIRST NUMBER UNIQUE] [RECORDS] [IN] [FILE] <i>view-name</i> [PASSW = <i>op-2</i>] [CIPHER = <i>op-3</i>] [WITH] [[LIMIT] (<i>op-4</i>)] <i>basic-search-criterion</i> [<AND OR> COUPLED [TO] [FILE] <i>view-name</i> [VIA <i>descriptor-1</i> <= EQ EQUAL [TO]> <i>descriptor-2</i>] [WITH] <i>basic-search-criterion</i>] [STARTING WITH ISN = <i>op-5</i>] [SORTED [BY] <i>descriptor-3</i> [DESCENDING]] [RETAIN AS <i>op-6</i>] [WHERE <i>logical-condition</i>] [IF NO [RECORDS] [FOUND] <ENTER <i>stmt-2</i>> <END-NOREC DO <i>stmt-2</i> DOEND>] <i>stmt-1</i> ... <END-FIND [LOOP]> 	V	S	<i>op-1, op-2, op-3, op-4, op-5, op-6</i>	<i>view-name</i>
FOR	<ul style="list-style-type: none"> Format 1: structured mode FOR <i>op-1</i> [[:]= EQ FROM] <i>op-2</i> [TO THRU] <i>op-3</i> [[STEP] <i>op-4</i>] <i>stmt</i> ... END-FOR 	V	S	<i>op-2, op-3, op-4</i>	<i>op-1</i>
	<ul style="list-style-type: none"> Format 2: reported mode FOR <i>op-1</i> [[:]= EQ FROM] <i>op-2</i> [TO THRU] <i>op-3</i> [[STEP] <i>op-4</i>] <i>stmt</i> ... [LOOP] 	V	S	<i>op-2, op-3, op-4</i>	<i>op-1</i>
FORMAT	<ul style="list-style-type: none"> Format FORMAT [(<i>rep</i>)] <i>parameter</i> ... 	V	S		
GET	<ul style="list-style-type: none"> Format GET [IN] [FILE] <i>view-name</i> [PASSWORD = <i>op-1</i>] [CIPHER = <i>op-2</i>] [RECORD] <<i>op-3</i> *ISN [(<i>r</i>)]> <i>op-4</i> ... 	V	S	<i>op-1, op-2, op-3, op-4</i>	<i>view-name</i>
GET SAME	<ul style="list-style-type: none"> Format GET SAME [(<i>r</i>)] [<i>op-1</i> ...] 	V	S		<i>op-1</i>
GET TRANSACTION DATA	<ul style="list-style-type: none"> Format GET TRANSACTION [DATA] <i>op-1</i> ... 	V	N		<i>op-1</i>

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
HISTOGRAM	<ul style="list-style-type: none"> Format 1: structured mode HISTOGRAM [ALL (op-1)] [VALUE] [IN] [FILE] view-name [PASSWORD = op-2] [[IN] <ASCENDING DESCENDING VARIABLE op-3> [SEQUENCE]] [VALUE] [FOR] [FIELD] op-4 [[[STARTING] [WITH FROM] [VALUES] op-5] [[THRU ENDING AT] op-6]] [WHERE logical-condition] stmt ... END-HISTOGRAM 	V	S	op-1, op-2, op-4, op-5, op-6	view-name
	<ul style="list-style-type: none"> Format 2: reported mode HISTOGRAM [ALL (op-1)] [VALUE] [IN] [FILE] view-name [PASSWORD = op-2] [[IN] <ASCENDING DESCENDING VARIABLE op-3> [SEQUENCE]] [VALUE] [FOR] [FIELD] op-4 [[[STARTING] [WITH FROM] [VALUES] op-5] [[THRU ENDING AT] op-6]] [WHERE logical-condition] stmt ... [LOOP] 	V	S	op-1, op-2, op-4, op-5, op-6	view-name
IF	<ul style="list-style-type: none"> Format 1: structured mode IF logical-condition [THEN] stmt ... [ELSE stmt ...] END-IF 	V	S	op-1	
	<ul style="list-style-type: none"> Format 2: reported mode IF logical-condition [THEN] <stmt ... DO stmt ... DOEND> [ELSE <stmt ... DO stmt ... DOEND>] 	V	S	op-1	
IF SELECTION	<ul style="list-style-type: none"> Format 1: structured mode IF SELECTION [NOT UNIQUE [IN [FIELDS]]] op-1 [THEN] stmt ... [ELSE stmt ...] END-IF 	V	S		
	<ul style="list-style-type: none"> Format 2: reported mode IF SELECTION [NOT UNIQUE [IN [FIELDS]]] op-1 [THEN] <stmt ... DO stmt ... DOEND> [ELSE <stmt ... DO stmt ... DOEND>] 	V	S		
IGNORE	<ul style="list-style-type: none"> Format IGNORE 	V	S		
INCLUDE	<ul style="list-style-type: none"> Format INCLUDE copycode-name [op-1 ...] 	V	S		

2-20 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
INPUT	<ul style="list-style-type: none"> • Format 1: dynamic screen layout specification INPUT [WINDOW = 'window-name'] [NO ERASE] [(statement-parameters)] [[WITH] TEXT [*op-1 op-2] [(attributes)] [,op-3 ...]] [MARK [POSITION op-4 [IN]] [FIELD] <op-1 *field-name>] [[AND] [SOUND] ALARM] [nX nT x / y] ['text' [(attributes)] 'c'(n) [(attributes)] '.' '=' /...] [*IN *OUT *OUTIN] op-1[(parameter)] ... 	V	S	op-1, op-2, op-3, op-4, op-5	op-6
	<ul style="list-style-type: none"> • Format 2: using predefined map layout INPUT [WINDOW = 'window-name'] [[WITH] TEXT [*op-1 op-2] [(attributes)] [,op-3 ...]] [MARK [POSITION op-4 [IN]] [FIELD] <op-1 *field-name>] [[AND] [SOUND] ALARM] [USING] MAP map-name [NO ERASE] [op-1 NO PARAMETER] 	V	S	op-1, op-2, op-3, op-4, op-5	op-6
INTERFACE	<ul style="list-style-type: none"> • Format INTERFACE interface-name [ID interface-GUID] [PROPERTY property-name [(format-length/array-definition)] [READONLY] [IS oper] END-PROPERTY] [PROPERTY method-name [IS subprogram-name] [PARAMETER <USING <local-data-area parameter-data-area> data-definition ...>] END-METHOD] END-INTERFACE See the "Definition types" row for details about 'data-definition' and 'array-definition' 	V	S		
LIMIT	<ul style="list-style-type: none"> • Format LIMIT n 	V	S		
LOOP	<ul style="list-style-type: none"> • Format [<u>CLOSE</u>] LOOP [(r)] 	V	N		
METHOD	<ul style="list-style-type: none"> • Format METHOD method-name OF [INTERFACE] interface-name IS subprogram-name END-METHOD 	V	S		
MOVE	<ul style="list-style-type: none"> • Format 1 MOVE [ROUNDED] op-1 [(parameter)] TO op-2 	V	S	op-1	op-2
	<ul style="list-style-type: none"> • Format 2 MOVE [ROUNDED] <op-1 <u>SUBSTRING</u> (op-1, op-3, op-4)> [(parameter)] TO <op-2 <u>SUBSTRING</u> (op-2, op-5, op-6)> 	V	S	op-1, op-3, op-4, op-5, op-6	op-2
	<ul style="list-style-type: none"> • Format 3 MOVE BY <[NAME] POSITION> op-1 TO op-2 	V	S	op-1	op-2

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
	<ul style="list-style-type: none"> Format 4 MOVE EDITED <i>op-1</i> TO <i>op-2</i> (EM = <i>value</i>) 	V	S	<i>op-1</i>	<i>op-2</i>
	<ul style="list-style-type: none"> Format 5 MOVE EDITED <i>op-1</i> (EM = <i>value</i>) TO <i>op-2</i> 	V	S	<i>op-1</i>	<i>op-2</i>
	<ul style="list-style-type: none"> Format 6 MOVE <LEFT RIGHT> [JUSTIFIED] <i>op-1</i> [(<i>parameter</i>)] TO <i>op-2</i> 	V	S	<i>op-1</i>	<i>op-2</i>
MOVE ALL	<ul style="list-style-type: none"> Format MOVE ALL <i>op-1</i> TO <i>op-2</i> [UNTIL <i>op-3</i>] 	V	N	<i>op-1</i> , <i>op-3</i>	<i>op-2</i>
MULTIPLY	<ul style="list-style-type: none"> Format 1 MULTIPLY [ROUNDED] <i>op-1</i> BY <i>op-2</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-1</i>
	<ul style="list-style-type: none"> Format 2 MULTIPLY [ROUNDED] <i>op-1</i> BY <i>op-2</i> GIVING <i>op-3</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-3</i>
NEWPAGE	<ul style="list-style-type: none"> Format NEWPAGE [(<i>rep</i>)] [EVEN [IF] TOP [OF] [PAGE] [IF WHEN] LESS [THAN] <i>op-1</i> [LINES] [LEFT]] [(WITH) TITLE] 	V	S	<i>op-1</i>	
OBTAIN	<ul style="list-style-type: none"> Format OBTAIN <i>op-1</i> ... 	V	S	<i>op-1</i>	
ON ERROR	<ul style="list-style-type: none"> Format 1: structured mode ON ERROR <i>stmt</i> ... END-ERROR 	V	N		
	<ul style="list-style-type: none"> Format 2: reporting mode ON ERROR <<i>stmt</i> ... DO <i>stmt</i> ... DOEND> 	V	N		
OPEN CON- VERSATION	<ul style="list-style-type: none"> Format OPEN CONVERSATION USING [SUBPROGRAMS] <i>op-1</i> ... 	V	S	<i>op-1</i>	

2-22 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
OPEN DIALOG	<ul style="list-style-type: none"> Format OPEN DIALOG <i>op-1</i> [USING] [PARENT] <i>op-2</i> [[GIVING] [DIALOG-ID] <i>op-3</i>] [WITH <<i>op-4</i> [(AD = <M O A>)] nX PARAMETERS <i>parameter-name</i> = <i>op-4</i> ... END-PARAMETERS]>] 	V	S	<i>op-1</i> <i>op-2</i> <i>op-4</i>	<i>op-3</i>
PASSW	<ul style="list-style-type: none"> Format PASSW = <i>op-1</i> 	V	S	<i>op-1</i>	
PERFORM	<ul style="list-style-type: none"> Format PERFORM <i>op-1</i> [<i>op-2</i> [(AD = <M O A>)] nX] ... 	V	S	<i>op-2</i>	<i>op-2</i>
PERFORM BREAK PROCESSING	<ul style="list-style-type: none"> Format PERFORM BREAK [PROCESSING] [(<i>r</i>)] AT BREAK <i>stmt</i> ... 	V	S		
PRINT	<ul style="list-style-type: none"> Format PRINT [(<i>rep</i>)] [NOTITLE] [NOHDR] [(<i>statement-parameters</i>)] [nX nT /] ... <'text' [(<i>attributes</i>)] 'c' (n) [(<i>attributes</i>)] [=] <i>op-1</i> [(<i>parameters</i>)]> ... 	V	S		
PROCESS	<ul style="list-style-type: none"> Format PROCESS <i>view-name</i> USING <i>op-1</i> = <i>op-2</i> GIVING <i>op-3</i> ... 	N	N		
PROCESS COMMAND	<ul style="list-style-type: none"> Format 1: structured mode PROCESS COMMAND ACTION <CLOSE <CHECK EXEC TEXT HELP> USING PROCESSOR-NAME = <i>op-1</i> COMMAND-LINE (<i>index</i> [:<i>index</i>]) = <i>op-2</i> GET USING PROCESSOR-NAME = <i>op-1</i> GETSET-FIELD-NAME = <i>op-3</i> SET USING PROCESSOR-NAME = <i>op-1</i> GETSET-FIELD-NAME = <i>op-3</i> GETSET-FIELD-VALUE = <i>op-4</i>> 	N	N		
	<ul style="list-style-type: none"> Format 2: reporting mode PROCESS COMMAND ACTION <CLOSE [GIVING NATURAL-ERROR] <CHECK EXEC TEXT HELP> USING PROCESSOR-NAME = <i>op-1</i> COMMAND-LINE (<i>index</i> [:<i>index</i>]) = <i>op-2</i> GIVING RESULT-FIELD (<i>index</i> [:<i>index</i>]) RETURN-CODE [NATURAL-ERROR] GET USING PROCESSOR-NAME = <i>op-1</i> GETSET-FIELD-NAME = <i>op-3</i> GIVING GETSET-FIELD-VALUE [NATURAL-ERROR] SET USING PROCESSOR-NAME = <i>op-1</i> GETSET-FIELD-NAME = <i>op-3</i> GETSET-FIELD-VALUE = <i>op-4</i> [GIVING NATURAL-ERROR]> 	N	N		

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
PROCESS GUI	<ul style="list-style-type: none"> Format PROCESS GUI ACTION <i>action-name</i> WITH <<i>op-1</i> ... <i>nX</i> ... PARAMETERS <i>parameter-name</i> = <i>op-1</i> ... END-PARAMETERS> [GIVING <i>op-2</i>] 	N	N		
PROCESS REPORTER	<ul style="list-style-type: none"> Format PROCESS REPORTER ACTION <INITIALIZE TERMINATE <OPEN CLOSE REPLACE-TABLE SET-PRINTER SET-PRINT-OPTIONS PRINT PREVIEW> WITH <<i>op-1</i> ... PARAMETERS <i>parameter-name</i> = <i>op-1</i> ... END-PARAMETERS> EDIT [WITH <<i>op-1</i> ... PARAMETERS <i>parameter-name</i> = <i>op-1</i> ... END-PARAMETERS>] [GIVING <i>op-2</i>] 	N	N		
PROPERTY	<ul style="list-style-type: none"> Format PROPERTY <i>property-name</i> OF [INTERFACE] <i>interface-name</i> IS <i>operand</i> END-PROPERTY 	V	S	<i>operand</i>	<i>property-name</i>
READ	<ul style="list-style-type: none"> Format 1: structured mode <READ BROWSE> [ALL (<i>op-1</i>)] [RECORDS] [IN] [FILE] <i>view-name</i> [PASSWORD = <i>op-2</i>] [CIPHER = <i>op-3</i>] [WITH REPOSITION] [[IN] [PHYSICAL] [ASCENDING DESCENDING VARIABLE <i>op-5</i>] [SEQUENCE]] [<BY WITH > <ISN <i>descriptor</i>> [<= EQ EQUAL [TO] [STARTING] FROM> <i>op-6</i>] [[THRU ENDING AT] <i>op-7</i>]] [STARTING WITH ISN = <i>op-4</i>] [WHERE <i>logical-condition</i>] <i>stmt</i> ... END-READ 	V	S	<i>op-1, op-2, op-3, op-4, op-5, op-6, op-7</i>	<i>view-name</i>
	<ul style="list-style-type: none"> Format 2: reporting mode <READ BROWSE> [ALL (<i>op-1</i>)] [RECORDS] [IN] [FILE] <i>view-name</i> [PASSWORD = <i>op-2</i>] [CIPHER = <i>op-3</i>] [WITH REPOSITION] [[IN] [PHYSICAL] [ASCENDING DESCENDING VARIABLE <i>op-5</i>] [SEQUENCE]] [<BY WITH > <ISN <i>descriptor</i>> [<= EQ EQUAL [TO] [STARTING] FROM> <i>op-6</i>] [[THRU ENDING AT] <i>op-7</i>]] [STARTING WITH ISN = <i>op-4</i>] [WHERE <i>logical-condition</i>] <i>stmt</i> ... [LOOP] 	V	S	<i>op-1, op-2, op-3, op-4, op-5, op-6, op-7</i>	<i>view-name</i>
READ WORK FILE	<ul style="list-style-type: none"> Format 1: structured mode READ WORK [FILE] <i>work-file-number</i> [ONCE] <RECORD <i>op-1</i> [AND] [SELECT] [OFFSET <i>n</i> FILLER <i>nX</i>] ... <i>op-2</i> ...> [GIVING LENGTH <i>op-3</i>] AT [END] [OF] [FILE] <i>stmt</i> ... END-ENDFILE <i>stmt</i>... END-WORK 	V	S		<i>op-1 op-3</i>
	<ul style="list-style-type: none"> Format 2: reporting mode READ WORK [FILE] <i>work-file-number</i> [ONCE] <RECORD <i>op-1</i> [AND] [SELECT] [OFFSET <i>n</i> FILLER <i>nX</i>] ... <i>op-2</i> ...> [GIVING LENGTH <i>op-3</i>] AT [END] [OF] [FILE] <<i>stmt</i> ... DO <i>stmt</i>... DOEND> <i>stmt</i> ... [LOOP] 	V	S		<i>op-1 op-3</i>
REDEFINE	<ul style="list-style-type: none"> Format REDEFINE <i>op-1</i> (<<i>nX</i> <i>op-2</i>> ...) ... 	V	S		

2-24 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
REDUCE	<ul style="list-style-type: none"> Format REDUCE [SIZE OF] DYNAMIC [VARIABLE] <i>op-1</i> TO <i>op-2</i> 	N	N		
REINPUT	<ul style="list-style-type: none"> Format REINPUT [FULL] [(<i>statement-parameters</i>)] <USING HELP [WITH] [TEXT] [*<i>op-1</i> <i>op-2</i>] [(<i>attributes</i>)] [,<i>op-3</i> ...]> [MARK [POSITION <i>op-4</i> [IN]] [FIELD] <<i>op-1</i> *<i>field-name</i>>] [[AND] [SOUND] ALARM] 	V	S	<i>op-1, op-2, op-3, op-4, op-5</i>	<i>op-5</i>
REJECT	<ul style="list-style-type: none"> Format REJECT [IF] <i>logical-condition</i> 	V	S		
RELEASE	<ul style="list-style-type: none"> Format RELEASE <STACK <u>SEIS</u> [<i>set-name</i> ...] VARIABLES> 	V	S		
REPEAT	<ul style="list-style-type: none"> Format 1 REPEAT <i>stmt</i> ... [<UNTIL WHILE> <i>logical-condition</i>] END-REPEAT 	V	S		
	<ul style="list-style-type: none"> Format 2 REPEAT [<UNTIL WHILE> <i>logical-condition</i>] <i>stmt</i> ... END-REPEAT 	V	S		
RESET	<ul style="list-style-type: none"> Format RESET [INITIAL] <i>op-1</i> 	V	S		<i>op-1</i>
RETRY	<ul style="list-style-type: none"> Format RETRY 	V	S		
RUN	<ul style="list-style-type: none"> Format RUN [REPEAT] <i>op-1</i> [<i>op-2</i> [(<i>parameter</i>)]] ... 	V	S	<i>op-1 op-2</i>	<i>op-2</i>
SEND EVENT	<ul style="list-style-type: none"> Format SEND EVENT <i>op-1</i> TO [DIALOG-ID] <<i>op-2</i> *DIALOG-ID> [WITH <<i>op-3</i> [(AD =<M O A>)] <i>nX</i>> ... USING [DIALOG] '<i>dialog-name</i>' WITH PARAMETERS <i>parameter-name</i> = <i>op-3</i> ... END-PARAMETERS] 	V	S	<i>op-1 op-2 op-3</i>	
SEND METHOD	<ul style="list-style-type: none"> Format SEND [METHOD] <i>op-1</i> TO [OBJECT] <i>op-2</i> [WITH <<i>op-3</i> [(AD =<M O A>)] <i>nX</i>> ...] [RETURN <i>op-4</i>] [GIVING <i>op-5</i>] 	V	S	<i>op-1 op-2 op-3</i>	<i>op-4 op-5</i>

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
SEPARATE	<ul style="list-style-type: none"> Format SEPARATE <op-1 SUBSTRING (op-1, op-2, op-3)> [LEFT [JUSTIFIED]] INTO op-4 ... [IGNORE REMAINDER op-5] [WITH [RETAINED] <[ANY] DELIMITERS INPUT DELIMITERS DELIMITERS op-6>] [[GIVING] NUMBER [IN] op-7] 	V	S	op-1 op-2 op-3 op-6	op-4 op-5 op-7
SET CONTROL	<ul style="list-style-type: none"> Format SET CONTROL op-1 	V	S	op-1	
SET GLOBALS	<ul style="list-style-type: none"> Format SET GLOBALS parameter ... 	V	S	parameter ...	
SET KEY	<ul style="list-style-type: none"> Format 1: affecting all keys SET KEY <ALL ON OFF COMMAND <ON OFF> NAMED OFF> 	V	S		
	<ul style="list-style-type: none"> Format 2: affecting individual keys SET KEY <PAn PFn CLR DYNAMIC op-1> [= <ON OFF DISABLED COMMAND <ON OFF>>] ... 	V	S	op-1	
	<ul style="list-style-type: none"> Format 3: affecting individual keys SET KEY <<PAn PFn CLR DYNAMIC op-1> [= <PGM op-2 HELP DATA op-3>] [NAMED <op-4 OFF>] ENTR NAMED <op-4 OFF>> 	V	S		
SET TIME	<ul style="list-style-type: none"> Format <SET TIME SETTIME> 	V	S		
SET WINDOW	<ul style="list-style-type: none"> Format SET WINDOW <'window-name' OFF> 	V	S		
SKIP	<ul style="list-style-type: none"> Format SKIP [(rep)] op-1 [LINES] 	V	S		

2-26 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
SORT	<ul style="list-style-type: none"> • Format 1: structured mode END-ALL [AND] SORT [THEM RECORDS] [BY] <i>op-1</i> [ASCENDING DESCENDING] ... <USING <i>op-2</i> USING KEYS> [GIVE <MAX MIN NMIN COUNT NCOUNT OLD AVER NAVER SUM TOTAL> ... [OF] <(<i>op-3</i> ...) <i>op-3</i> ...> [(NL=<i>nn</i>)] ...] <i>stmt</i> ... END-SORT 	V	S	<i>op-1</i> <i>op-2</i>	
	<ul style="list-style-type: none"> • Format 2: reporting mode SORT [THEM RECORDS] [BY] <i>op-1</i> [ASCENDING DESCENDING] ... <USING <i>op-2</i> USING KEYS> [GIVE <MAX MIN NMIN COUNT NCOUNT OLD AVER NAVER SUM TOTAL> ... [OF] <(<i>op-3</i> ...) <i>op-3</i> ...> [(NL=<i>nn</i>)] ...] <i>stmt</i> ... 	V	S	<i>op-1</i> <i>op-2</i>	
STACK	<ul style="list-style-type: none"> • Format STACK [TOP] <COMMAND <i>op-1</i> [<i>op-2</i> [(<i>parameter</i>)] ... [DATA] [FORMATTED] <i>op-1</i> [(<i>parameter</i>)] ...> 	V	S	<i>op-1</i> <i>op-2</i>	
STOP	<ul style="list-style-type: none"> • Format STOP 	V	S		
STORE	<ul style="list-style-type: none"> • Format 1: structured mode STORE [RECORD] [IN] [FILE] <i>view-name</i> [PASSWORD = <i>op-1</i>] [CIPHER = <i>op-2</i>] [[USING GIVING] NUMBER <i>op-3</i>] [(<i>r</i>)] 	V	S	<i>view-name</i> , <i>op-1</i> , <i>op-2</i> , <i>op-3</i>	
	<ul style="list-style-type: none"> • Format 2: reporting mode STORE [RECORD] [IN] [FILE] <i>view-name</i> [PASSWORD = <i>op-1</i>] [CIPHER = <i>op-2</i>] [[USING GIVING] NUMBER <i>op-3</i>] [(<i>r</i>)] <[USING] SAME [RECORD] [AS] [STATEMENT [(<i>r</i>)] [SET WITH] [<i>op-4</i> = <i>op-5</i>] ...> 	V	S	<i>view-name</i> , <i>op-1</i> , <i>op-2</i> , <i>op-3</i>	
SUBTRACT	<ul style="list-style-type: none"> • Format 1 SUBTRACT [ROUNDED] <i>op-1</i> ... FROM <i>op-2</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-2</i>
	<ul style="list-style-type: none"> • Format 2 SUBTRACT [ROUNDED] <i>op-1</i> ... FROM <i>op-2</i> GIVING <i>op-3</i> 	V	S	<i>op-1</i> <i>op-2</i>	<i>op-3</i>
SUSPEND IDENTICAL SUPPRESS	<ul style="list-style-type: none"> • Format SUSPEND IDENTICAL [SUPPRESS] [(<i>rep</i>)] 	V	S		
TERMINATE	<ul style="list-style-type: none"> • Format TERMINATE [<i>op-1</i> [<i>op-2</i>]] 	V	S	<i>op-1</i> <i>op-2</i>	

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
UPDATE	<ul style="list-style-type: none"> Format 1: structured mode UPDATE [RECORD] [IN] [STATEMENT] [(r)] 	V	S		
	<ul style="list-style-type: none"> Format 2: reporting mode UPDATE [RECORD] [IN] [STATEMENT] [(r)] [SET WITH USING] <SAME [RECORD] op-1 = op-2 ...> 	V	S	op-1 op-2	
WRITE	<ul style="list-style-type: none"> Format 1: dynamic formatting WRITE [(rep)] [NOTITLE] [NOHDR] [(statement-parameters)] [nX nT x/y T*field-name P*field-name /] ... <text' [(attributes)] 'c' (n) [(attributes)] [=] op-1 [(parameters)]> ... 	V	S	op-1	
	<ul style="list-style-type: none"> Format 2: using predefined map WRITE [(rep)] [NOTITLE] [NOHDR] [USING] <FORM MAP> op-1 [op-2 ...] 	V	S	op-1 op-2	
WRITE TITLE	<ul style="list-style-type: none"> Format WRITE [(rep)] TITLE [LEFT [JUSTIFIED]] [UNDERLINED] [(statement-parameters)] [nX nT x/y] ... <text' [(attributes)] 'c' (n) [(attributes)] [=] op-1 [(parameters)]> ... [SKIP op-2 [LINES]] 	V	S	op-1 op-2	
WRITE TRAILER	<ul style="list-style-type: none"> Format WRITE [(rep)] TRAILER [LEFT [JUSTIFIED]] [UNDERLINED] [(statement-parameters)] [nX nT x/y] ... <text' [(attributes)] 'c' (n) [(attributes)] [=] op-1 [(parameters)]> ... [SKIP op-2 [LINES]] 	V	N		
WRITE WORK FILE	<ul style="list-style-type: none"> Format WRITE WORK [FILE] work-file-number [VARIABLE] op-1 ... 	V	S	op-1	

2-28 Supported Natural Statements
Supported Natural 3.1.3 for Mainframes / 4.1.2 for Windows statements

SQL Statements

Statement	Parser	CE
CALLDBPROC	V	S
COMMIT	V	S
DELETE	V	S
INSERT	V	S
PROCESS SQL	N	N
READ RESULT SET	V	S
ROLLBACK	V	S
SELECT	V	S
UPDATE	V	S

Supported PL/I Statements



This chapter describes the PL/I statements supported by ATW for component extraction. For a list of supported legacy versions, refer to the *Release Notes*.

Key to tables

The tables below contain detailed information on ATW component extraction support. The first column of each table presents PL/I keywords. Other columns provide the following information:

- **Parser** — ATW Parser. The possible values in this column are:
 - (V)erified — parsed and prepared for further processing. Possibly is supported by other ATW tools. Does not initiate any errors or warnings during the verification phase.
 - (V)erification (O)nly — parsed and skipped immediately. Further processing is not possible, i.e. not supported by ATW tools. Verification warning is initiated.

3-2 Supported PL/I Statements
Key to tables

- (N)ot Verified — if the ATW parser finds such an entity it stops and verification is considered as erroneous.
- **CE** — Component Extraction. Component Extraction (S)upport means that the entity can be correctly sliced. If the entity is (N)ot supported and is represented in the slicing code the obtained program may not work correctly. The ‘Gen.’ column gives information on ATW support for:
 - Structure-Based Component Extraction
 - Domain-Based Component Extraction.

Supported IBM Visual Age PL/I 2.0 statements

Problem data types

PL/I language construction	Parser	CE
BIN FIXED	V	S
BIN FLOAT	V	S
DEC FIXED	V	S
DEC FLOAT	V	S
CHAR	V	S
BIT	V	S
PIC	V	N
DATE	N	N
GRAPHIC	VO	N
COMPLEX	VO	N
VARYING	VO	N

Program-control data types

PL/I language construction	Parser	CE
LABEL attribute	V	N

3-4 Supported PL/I Statements
Supported IBM Visual Age PL/I 2.0 statements

Label data	PS ^a	N
GENERIC attribute	PS ^b	S
Entry constants (functions and procedures)	V	S
Entry variables	V	N
OPTIONS option	V	S
Returns option and attribute	V	S

- a. The in-place initialization of an array of labels is not supported.
- b. Incorrect work in the case of using ENTRY, POINTER in the generic-descriptor

Aggregate types and attributes

PL/I language construction	Parser	CE
Arrays	V	N
Structures	V	N
Structures with array field	V	N
Array of structures	V	N
Select from structure of arrays	V	N
LIKE attribute	V	S

Expressions and references

PL/I language construction		Parser	CE	
Pointer operations	+	VO	N	
	-	VO	N	
	compare	V	N	
	built-ins	VO	N	
Arithmetic operations	+	Unary	S	
		Binary	S	
	-	Unary	S	
		Binary	S	
	*		V	S
	/		V	S
	**		V	S
	+=		N	N
	-=		N	N
	*=		N	N
	/=		N	N
	**=		N	N

3-6 Supported PL/I Statements
Supported IBM Visual Age PL/I 2.0 statements

Bit operations		V	S
	&	V	S
	^	V	S
	=	N	N
	&=	N	N
Comparison operations	<	V	S
	^<	V	S
	<=	V	S
	=	V	S
	^=	V	S
	>=	V	S
	>	V	S
	^>	V	S
Concatenation		V	S
	=	N	N

Array expressions	+	Unary	V	N
		Binary	V	N
	-	Unary	V	N
		Binary	V	N
	*		V	N
	/		V	N
	**		V	N
			V	N
	&		V	N
	^		V	N
	<		V	N
	^<		V	N
	<=		V	N
	=		V	N
	^=		V	N
	>=		V	N
Structure expressions	=		V	N
	= by name		V	N

3-8 Supported PL/I Statements
Supported IBM Visual Age PL/I 2.0 statements

Statements

PL/I language construction	Parser	CE
ALLOCATE	V	N
Assignment	V	S
BEGIN	V	S
CALL	V	PS ^a
CLOSE	VO	S
DECLARE	V	S
DELAY	VO	S
DELETE	VO	S
DISPLAY	V	S
DO (type 1: DO;)	V	S
DO (type 2: DO WHILE/UNTIL)	V	S
DO (type 3: DO <.>=.. TO ... BY ... WHILE .. UNTIL ...REPEAT ...)	V	S
DO (type 4: DO <.>=.,...,.. TO.. BY., WHILE ..., UNTIL .., REPEAT ...)	V	S
END	V	S

ENTRY	V	S
EXIT	VO	S
FETCH	VO	S
FLUSH	VO	S
FORMAT	VO	S
FREE	V	N
GET	V	S
GOTO	V	PS ^b
IF	V	S
LEAVE	V	S
LOCATE	VO	N
null statement	V	S
ON	V	N
OPEN	VO	S
OTHERWISE	V	S
PROCEDURE	V	S
PUT	V	S
READ	V	S
RELEASE	VO	S
RETURN	V	S

3-10 Supported PL/I Statements
Supported IBM Visual Age PL/I 2.0 statements

REVERT	VO	N
REWRITE	V	S
SELECT	V	S
SIGNAL	V	N
STOP	V	S
WHEN	V	S
WRITE	V	S

- a. ENTRY variables are not supported.
- b. LABEL variables are not supported.

Built-in functions and pseudovariables

PL/I language construction	Parser	CE
ABS	V	S
ACOS	V	S
ADD	V	S
ADDR	V	S
ALL	V	S
ALLOCATION	VO	S
ANY	V	S

ASIN	V	S
ATAN	V	S
ATAND	V	S
ATANH	V	S
BINARY	V	S
BINARYVALUE	VO	S
BIT	V	S
BOOL	V	S
CEIL	V	S
CHARACTER	V	S
COMPLEX	VO	S
CONJG	VO	S
COS	V	S
COSD	V	S
COSH	V	S
COTAN	VO	S
COTAND	VO	S
COUNT	VO	S
CURRENTSTORAGE	V	S
DATAFIELD	V	S

3-12 Supported PL/I Statements
Supported IBM Visual Age PL/I 2.0 statements

DATE	V	S
DATETIME	V	S
DECIMAL	V	S
DIMENSION	V	S
DIVIDE	V	S
EMPTY	VO	S
ENTRYADDR	VO	S
ERF	V	S
ERFC	V	S
EXP	V	S
FIXED	V	S
FLOAT	V	S
FLOOR	V	S
GRAPHIC	VO	S
HBOUND	V	S
HIGH	V	S
IMAG	VO	S
INDEX	V	S
LBOUND	V	S
LENGTH	V	S

LINENO	VO	S
LOG	V	S
LOG2	V	S
LOG10	V	S
LOW	V	S
MAX	V	S
MIN	V	S
MOD	V	S
MPSTR	V	S
MULTIPLY	V	S
NULL	V	S
OFFSET	VO	S
ONCHAR	V	S
ONCODE	V	S
ONCOUNT	VO	S
ONFILE	VO	S
ONKEY	VO	S
ONLOC	V	S
ONSOURCE	V	S
ONSUBCODE	VO	S

3-14 Supported PL/I Statements
Supported IBM Visual Age PL/I 2.0 statements

PLIDUMP	VO	S
PLIRETC	VO	S
PLIRETV	V	S
PLISRTA	VO	S
PLISRTB	VO	S
PLISRTC	VO	S
PLISRTD	VO	S
POINTER	VO	S
POINTERADD	VO	S
POINTVALUE	VO	S
PRECISION	V	S
PRESENT	VO	S
PROD	V	S
REAL	VO	S
REPEAT	V	S
ROUND	V	S
SAMEKEY	VO	S
SIGN	V	S
SIN	V	S
SIND	V	S

SINH	V	S
SQRT	V	S
STORAGE	V	S
STRING	V	S
SUBSTR	V	S
SUM	V	S
SYSNULL	V	S
SYSTEM	VO	S
TAN	V	S
TAND	V	S
TANH	V	S
TIME	V	S
TRANSLATE	V	S
TRUNC	V	S
UNSPEC	V	S
VERIFY	V	S

3-16 Supported PL/I Statements
Supported IBM Visual Age PL/I 2.0 statements

Supported SQL, CICS, and IMS Statements



This chapter describes SQL, CICS, and IMS statements supported by ATW for component extraction. For a list of supported legacy versions, refer to the *Release Notes*.

Supported SQL Statements

Comparisons

Note: If *expression* is specified in ‘Read Variables’ Column, then all host-variables contained in this expression are specified in this column.

Predicates	Format	Parser	CE	
			‘Read’ variables	‘Written’ variables
BASIC	<ul style="list-style-type: none"> Format <i>expr-1</i> <= <> > < >= <= > <i>expr-2</i> 	V	<i>expr-1, expr-2</i>	
QUANTIFIED	<ul style="list-style-type: none"> Format 1 <i>expr</i> <= <> > < >= <= > < SOME ANY ALL > (fullselect) 	V	<i>expr</i>	
	<ul style="list-style-type: none"> Format 2 (<i>expr-1, expr-2,...</i>) <= <> > < >= <= > < SOME ANY ALL > (fullselect1) 	N	<i>expr-1, expr-2,...</i>	
BETWEEN	<ul style="list-style-type: none"> Format <i>expr-1</i> [NOT] BETWEEN <i>expr-2</i> AND <i>expr-3</i> 	V	<i>expr-1, expr-2, expr-3</i>	
EXISTS	<ul style="list-style-type: none"> Format [NOT] EXISTS (fullselect) 	V		
IN	<ul style="list-style-type: none"> Format 1- <i>expr-1</i> [NOT] IN (<i>expr-2, expr-3,...</i>) 	V	<i>expr-1, expr-2, expr3, ...</i>	
	<ul style="list-style-type: none"> (Format 2 (<i>expr-1, expr-2,...</i>) [NOT] IN (<i>expr-3, expr-4,...</i>) 	N	<i>expr-1, expr-2, expr-3, expr-4</i>	

Predicates	Format	Parser	CE	
			'Read' variables	'Written' variables
LIKE	<ul style="list-style-type: none"> Format 1 <i>< constant host-variable column-name > [NOT] LIKE pattern-expr</i> 	V	<i>host-variable pattern-expr</i>	
	<ul style="list-style-type: none"> Format 2 <i>special register [NOT] LIKE pattern-expr</i> 	V*	<i>pattern-expr</i>	
	<ul style="list-style-type: none"> Format 3 <i>scalar function [NOT] LIKE pattern-expr</i> 	V* *	<i>pattern-expr</i>	
	<ul style="list-style-type: none"> Format 4 <i>large object locator [NOT] LIKE pattern-expr</i> 	N		
	<ul style="list-style-type: none"> Format 5 <i>expr LIKE scalar function whose operands are constant / special register / host variable</i> 	V*	<i>expr, host variable</i>	
	<ul style="list-style-type: none"> Format 6 <i>expr LIKE special register</i> 	V*	<i>expr</i>	
	<ul style="list-style-type: none"> Format 7 <i>expr-1 LIKE < constant host-variable expr-2 concatenating any of the above ></i> 	V	<i>expr-1, host variable, expr-2</i>	
NULL	<ul style="list-style-type: none"> Format <i>expr IS [NOT] NULL</i> 	V	<i>expr</i>	

* Works only with supported special registers

** Works only with supported scalar functions

4-4 Supported SQL, CICS, and IMS Statements
Supported SQL Statements

Essential Statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Writ-ten' variables
BEGIN DECLARE SECTION	<ul style="list-style-type: none"> Format BEGIN DECLARE SECTION 	V	S		
CLOSE CURSOR	<ul style="list-style-type: none"> Format 1 CLOSE <i>cursor-name</i> 	V	S		
	<ul style="list-style-type: none"> Format 2 CLOSE <i>cursor-name</i> WITH RELEASE 	N	N		
COMMIT	<ul style="list-style-type: none"> Format COMMIT 	V	S		
CONNECT	<ul style="list-style-type: none"> Format CONNECT TO <<i>server-name</i> <i>host-variable-1</i>> USER <<i>authorization-name</i> <i>host-variable-2</i>> USING <<i>password</i> <i>host-variable-3</i>> 	N	N		
DECLARE CURSOR	<ul style="list-style-type: none"> Format DECLARE <i>cursor-name</i> CURSOR FOR <<i>select-stmt</i> <i>stmt-name</i>> 	V	S		
DELETE	<ul style="list-style-type: none"> Format 1 DELETE FROM <i>table-name</i> WHERE <i>search-condition</i> 	V	S	<i>search-condition</i>	
	<ul style="list-style-type: none"> Format 2 DELETE FROM <i>table-name</i> WHERE CURRENT OF <i>cursor-name</i> 	V	S		
DISCONNECT	<ul style="list-style-type: none"> Format DISCONNECT <i>server-name/host-variable/ALL/CURRENT</i> 	N	N		
END DECLARE SECTION	<ul style="list-style-type: none"> Format END DECLARE SECTION 	V	S		

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Writ-ten' variables
FETCH	<ul style="list-style-type: none"> Format 1 FETCH <i>cursor-name</i> INTO <i>host-variable</i> 	V	S		<i>host-variable</i>
	<ul style="list-style-type: none"> Format 2 FETCH <i>cursor-name</i> USING DESCRIPTOR <i>descriptor-name</i> 	N	N		
INCLUDE	<ul style="list-style-type: none"> Format 1 INCLUDE SQLCA 	V	S		
	<ul style="list-style-type: none"> Format 2 INCLUDE SQLDA 	V	S		
INSERT	<ul style="list-style-type: none"> Format 1 INSERT INTO <i>table-name</i> (<i>col-name1</i>, <i>col-name2</i>,...) VALUES [< <i>expr-11</i> NULL >] 	V	S	<i>expr-1</i>	
	<ul style="list-style-type: none"> Format 2 INSERT INTO <i>table-name1</i> SELECT <i>expr</i> FROM <i>table-name2</i> 	V	S	<i>expr</i>	
	<ul style="list-style-type: none"> Format 3 INSERT INTO <i>table-name</i> (<i>col-name1</i>, <i>col-name2</i>,...) VALUES [< <i>expr</i> NULL ,...>] WITH <i>common-table-expr</i> 	N	N		
OPEN CURSOR	<ul style="list-style-type: none"> Format 1 OPEN <i>cursor-name</i> 	V	S		
	<ul style="list-style-type: none"> Format 2 OPEN <i>cursor-name</i> USING <i>host-variable1</i>, <i>host-variable2</i>,... 	V	S		
	<ul style="list-style-type: none"> Format 3 OPEN <i>cursor-name</i> USING DESCRIPTOR <i>descriptor-name</i> 	N	N		
ROLLBACK	<ul style="list-style-type: none"> Format ROLLBACK 	V	S		

4-6 Supported SQL, CICS, and IMS Statements
Supported SQL Statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
SELECT	<ul style="list-style-type: none"> Format 1 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-ref WHERE expr 	V	S	expr	host-variable
	<ul style="list-style-type: none"> Format 2 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-ref AS correlation-name WHERE expr 	V	S	expr	host-variable
	<ul style="list-style-type: none"> Format 3 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-ref WHERE expr GROUP BY grouping-expr 	V	S	expr	host-variable
	<ul style="list-style-type: none"> Format 4 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-ref WHERE expr ORDER BY column-name1, ... < ASC DESC > 	V	S	expr	host-variable
	<ul style="list-style-type: none"> Format 5 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-ref WHERE [NOT] < predicate search-condition > 	V	S	search-condition predicate	host-variable
	<ul style="list-style-type: none"> Format 6 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-ref WHERE < predicate-1 search-condition > < AND OR > < predicate-2 search-condition2 > 	V	S	search conditions	host-variable
	<ul style="list-style-type: none"> Format 7 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-ref WHERE expr-1 < = < > < > <= >= > expr-2 	V	S	expr-1 expr-2	host-variable

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
SELECT	<ul style="list-style-type: none"> Format 8 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-ref</i> WHERE <i>expr-1</i> BETWEEN <i>expr-2</i> AND <i>expr-3</i> 	V	S	<i>expr-1</i> <i>expr-2</i> <i>expr-3</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 9 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-ref</i> WHERE <i>expr</i> [NOT] IN (<i>expr-2</i>, ...) 	V	S	<i>expr</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 10 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-ref</i> WHERE <i>match-expr</i> [NOT] LIKE <i>pattern-expr</i> 	V	S	<i>match-expr</i> <i>pattern-expr</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 11 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-ref</i> WHERE <i>expr IS</i> [NOT] NULL 	V	S	<i>expr</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 12 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-def</i> WHERE <i>expr</i> [< UNION UNION ALL >] [<i>subselect</i> <i>fullselect</i>] 	V	S	<i>expr</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 13 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-def</i> WHERE <i>expr</i> [< EXCEPT EXCEPT ALL >] [< <i>subselect</i> <i>fullselect</i> >] 	V	S	<i>expr</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 14 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-def</i> WHERE <i>expr</i> [< INSPECT INSPECT ALL >] [< <i>subselect</i> <i>fullselect</i> >] 	V	S	<i>expr</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 15 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-def</i> WHERE <i>expr</i> [< INNER LEFT RIGHT FULL >] JOIN <i>expr-2</i> ON 	V	S	<i>expr</i> <i>expr-2</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 16 SELECT < <i>table-name</i> <i>view-name</i> <i>correlation-name</i> <i>column-name</i> > INTO <i>host-variable</i> FROM <i>table-def</i> WHERE <i>expr</i> FOR [<UPDATE READ ONLY OPTIMIZE>] FOR 	V	S	<i>expr</i>	<i>host-variable</i>
	<ul style="list-style-type: none"> Format 17 SELECT <i>column-name</i> AS <i>new-column-name</i> INTO <i>host-variable</i> FROM <i>table-def</i> WHERE <i>expr</i> 	N	N		

4-8 Supported SQL, CICS, and IMS Statements
Supported SQL Statements

Statement	Format	Parser	CE		
			Gen.	'Read' variables	'Written' variables
SELECT	<ul style="list-style-type: none"> Format 19 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-def WHERE expr VALUES [<expression NULL>] 	N	N		
	<ul style="list-style-type: none"> Format 20 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-def WHERE expr ORDER BY sort-key-expression 	N	N		
	<ul style="list-style-type: none"> Format 21 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-def WHERE expr < = > < <> >= > < SOME ANY ALL> 	N	N		
	<ul style="list-style-type: none"> Format 22 SELECT < table-name view-name correlation-name column-name > INTO host-variable FROM table-def WHERE EXISTS expr 	N	N		
UPDATE	<ul style="list-style-type: none"> Format 1 UPDATE table-name SET column-name1 = < expression1 NULL, ... > WHERE search-condition 	V	S	search-condition expr 1	
	<ul style="list-style-type: none"> Format 2 UPDATE table-name WHERE CURRENT OF cursor-name 	V	S		
	<ul style="list-style-type: none"> Format 3 UPDATE table-name SET (column-name1, column-name2,...) = (<expression1 NULL >, <expression2 NULL>, ...) WHERE search-condition 	N	N		
WHENEVER	<ul style="list-style-type: none"> Format WHENEVER < Not found SQLerror SQLwarning> <CONTINUE GOTO host-label> 	V	S		

Supported CICS command statements

Statement	Parser	CE
ADDRESS	V	S
ABEND	V	S
ALLOCATE	V	S
ASKTIME	V	S
ASSIGN	V	S
CANCEL	V	S
CONNECT PROCESS	V	S
CONVERSE	V	S
DELETE	V	S
DELETEQ TD	V	S
DELETEQ TS	V	S
DUMP	V	S
ENDBR	V	S
FORMATTIME	V	S
FREE	V	S
FREEMAIN	V	S
GETMAIN	V	S
HANDLE ABEND	V	S
HANDLE AID	V	S
HANDLE CONDITION	V	S
IGNORE CONDITION	V	S

4-10 Supported SQL, CICS, and IMS Statements
Supported CICS command statements

Statement	Parser	CE
INQUIRE	V	S
JOURNAL	V	S
LINK	V	S
LOAD	V	S
POP HANDLE	V	S
PURGE MESSAGE	V	S
PUSH HANDLE	V	S
READ	V	S
READNEXT	V	S
READPREV	V	S
READQ TD	V	S
READQ TS	V	S
RECEIVE	V	S
RECEIVE MAP	V	S
RECEIVE MAPSET	V	S
RELEASE	V	S
RESETBR	V	S
RETRIEVE	V	S
RETURN	V	S
REWRITE	V	S
ROUTE	V	S
SEND	V	S
SEND MAP	V	S

Statement	Parser	CE
SEND MAPSET	V	S
SEND TEXT	V	S
SET	V	S
SIGNOFF	V	S
SIGNON	V	S
SPOOLCLOSE	V	S
SPOOLOPEN INPUT	V	S
SPOOLOPEN OUTPUT	V	S
SPOOLREAD	V	S
SPOOLWRITE	V	S
START	V	S
STARTBR	V	S
SYNCPOINT	V	S
TRACE	V	S
UNLOCK	V	S
WAIT JOURNAL	V	S
WRITE	V	S
WRITEQ TD	V	S
WRITEQ TS	V	S
XCTL	V	S

4-12 Supported SQL, CICS, and IMS Statements
Supported IMS/Exec DLI statements

Supported IMS/Exec DLI statements

Statement	Parser	CE
GET (GU, GN, GNP)	V	S
TERM	V	S
SCHEDULE	V	S
ISRT	V	S
REPL	V	S
DLET	V	S

Supported RDMS 2200 SQL Statements

Statement	Format	Parser	CE
ALLOCATE CURSOR	<ul style="list-style-type: none"> Format <i>ALLOCATE cursor-name CURSOR [FOR RETENTION] FOR statement-name</i> 	VO	N
ALTER TABLE	<ul style="list-style-type: none"> Format <i>ALTER TABLE table-specification edit-specification-list</i> 	V	N
BEGIN DECLARE	<ul style="list-style-type: none"> Format BEGIN DECLARE SECTION 	VO	N
BEGIN THREAD	<ul style="list-style-type: none"> Format <i>BEGIN THREAD [thread-name] FOR [APPLICATION] application-name [READ RETRIEVE UPDATE [(DEFERRED QUICKLOOKS COMMANDLOOKS NONE)]] [UDSMMSG UDSMESSAGE UDSMESSAGES]</i> 	VO	N
CALL	<ul style="list-style-type: none"> Format <i>CALL [schema-name.]routine-name (parameter, ...)</i> 	VO	N

Statement	Format	Parser	CE
CLOSE	<ul style="list-style-type: none"> Format <CLOSE RELEASE [CURSOR]> <i>cursor-name</i> 	V	N
COMMIT	<ul style="list-style-type: none"> Format COMMIT [WORK THREAD] [ADVANCE TERMINATE] 	V	N
CREATE INDEX	<ul style="list-style-type: none"> Format CREATE INDEX <i>index-name</i> ON <i>table-specification</i> (<i>sort-column-list</i>) [IN [<i>schema-name</i>.]<i>storage-area-name</i>] 	V	N
CREATE SCHEMA	<ul style="list-style-type: none"> Format CREATE SCHEMA [<i>schema-name</i>] AUTHORIZATION <i>authorization-id</i> [<i>table-definition</i> <i>view-definition</i> <i>privilege-definition</i>] 	VO	N
CREATE TABLE	<ul style="list-style-type: none"> Format CREATE TABLE <i>table-specification</i> ([<i>storage-area-specification</i>,] <i>column-definition</i> [, <<i>column-definition</i> <i>table-constraint-specification</i> <i>storage-area-specification</i> <i>index-specification</i>>] ...) 	V	N
CREATE VIEW	<ul style="list-style-type: none"> Format CREATE VIEW <i>view-specification</i> [(<i>column-name</i> ...)] AS <i>query-specification</i> [WITH CHECK OPTION] [WITH ACCESS CONTROL] 	V	N
DEBUG	<ul style="list-style-type: none"> Format DEBUG <DUMP <i>dump-option</i> NODUMB <i>dumb-option</i> STUB NOSTUB PARSE EXECUTE> <i>where dump-option is</i> <COMMAND MODULE DATA PACKET REQUEST PACKET RDMCA RSP PACKET WORK AREA ALL SIZES INPUT ECHO> 	V	N
DECLARE CURSOR	<ul style="list-style-type: none"> Format <DECLARE <i>cursor-name</i> CURSOR DEFINE CURSOR <i>cursor-name</i>> [FOR <SEQUENTIAL RANDOM DIRECT> [ACCESS]] [[FOR] RETENTION] [FOR] <i>query-expression</i> [ORDER BY <i>sort-specification-list</i> FOR UPDATE OF <i>column-name-list</i>] [WITH DESCRIPTION] 	V	N
DELETE	<ul style="list-style-type: none"> Format DELETE [FROM] <i>table-specification</i> [ALL [ROWS] WHERE <<i>boolean-expr</i> CURRENT OF <i>cursor-name</i>>] 	V	N
DROP CURSOR	<ul style="list-style-type: none"> Format DROP CURSOR <i>cursor-name</i> 	VO	N
DROP INDEX	<ul style="list-style-type: none"> Format DROP INDEX <i>index-name-list</i> FROM [TABLE] <i>table-specification</i> 	VO	N

4-14 Supported SQL, CICS, and IMS Statements
Supported RDMS 2200 SQL Statements

Statement	Format	Parser	CE
DROP PROCEDURE/FUNCTION	<ul style="list-style-type: none"> Format DROP <PROCEDURE FUNCTION> [<i>schema-name</i>.]<i>routine-name</i> <CASCADE RESTRICT> 	VO	N
DROP TABLE	<ul style="list-style-type: none"> Format DROP [PERMANENT] TABLE <i>table-list</i> 	V	N
DROP VIEW	<ul style="list-style-type: none"> Format DRO VIEW [<i>schema-name</i>.]<i>routine-name</i> ... 	V	N
END DECLARE	<ul style="list-style-type: none"> Format END DECLARE SECTION 	VO	N
END THREAD	<ul style="list-style-type: none"> Format END THREAD [ADVANCE TERMINATE] 	VO	N
EXECUTE	<ul style="list-style-type: none"> Format EXECUTE <i>statement-name</i> [USING <i>host-variable-list</i>] 	V	N
EXECUTE IMMEDIATE	<ul style="list-style-type: none"> Format EXECUTE IMMEDIATE <:<i>embedded-variable</i> <i>string-literal</i>> 	VO	N
EXPLAIN	<ul style="list-style-type: none"> Format EXPLAIN <i>cursor-name</i> 	VO	N
FETCH	<ul style="list-style-type: none"> Format FETCH [NEXT FIRST PRIOR LAST CURRENT] [FROM] <i>cursor-name</i> INTO <i>variable-specification-list</i> 	V	N
FETCH NEXT	<ul style="list-style-type: none"> Format FETCH NEXT <i>number-of-rows</i> [FROM] <i>cursor-name</i> INTO <i>variable-specification-list</i> 	V	N
FUNCTION	<ul style="list-style-type: none"> Format FUNCTION [<i>schema-name</i>.]<i>routine-name</i> (<i>parameter</i> ...) RETURNS <i>data-type</i> <i>stored-procedure-statement</i> 	VO	N
GET DESCRIPTION	<ul style="list-style-type: none"> Format GET DESCRIPTION INTO <i>placeholder-list</i> 	VO	N

Statement	Format	Parser	CE
GETERROR	<ul style="list-style-type: none"> Format GETERROR INTO <i>variable-list</i> 	VO	N
GET PARAMETERS	<ul style="list-style-type: none"> Format GET PARAMETERS FOR [<i>schema-name</i>].<i>routine-name</i> INTO <i>placeholder-list</i> 	VO	N
GRANT	<ul style="list-style-type: none"> Format GRANT <ALL [PRIVILEGES] <i>privilege-list</i>> ON [TABLE] <i>table-specification</i> TO <i>user-id-list</i> [WITH GRANT OPTION] 	V	N
INSERT	<ul style="list-style-type: none"> Format INSERT [INTO] <i>table-specification</i> <[(<i>column-name-list</i>)] <VALUES (<i>value-list</i>) <i>query-specification</i>> COLUMNS (<i>column-and-value</i>)> 	V	N
LEVEL	<ul style="list-style-type: none"> Format LEVEL [RSA RDMS] INTO <i>placeholder</i> 	VO	N
LOCATE	<ul style="list-style-type: none"> Format LOCATE <i>cursor-name</i> ON <i>table-specification</i> USING VALUES (<i>primary-key-value</i>) 	VO	N
LOCK	<ul style="list-style-type: none"> Format LOCK [TABLE] <i>table-specification-list</i> [IN <i>lock-specification</i> [MODE]] [ON CONFLICT <RETURN QUEUE>] 	VO	N
OPEN	<ul style="list-style-type: none"> Format <OPEN READY [CURSOR]> <i>cursor-name</i> [USING <i>variable-list</i>] 	V	N
PREPARE	<ul style="list-style-type: none"> Format PREPARE <i>statement-name</i> FROM <:<i>embedded-variable</i> <i>string-literal</i>> 	V	N
PROCEDURE	<ul style="list-style-type: none"> Format PROCEDURE [<i>schema-name</i>].<i>routine-name</i> (<i>parameter</i> ...) <i>stored-procedure-statement</i> 	VO	N
REVOKE	<ul style="list-style-type: none"> Format REVOKE <ALL [PRIVILEGES] <i>privilege-list</i>> ON <i>table-specification</i> FROM <i>user-id-list</i> 	V	N
ROLLBACK	<ul style="list-style-type: none"> Format <ROLLBACK [WORK] OMIT THREAD> [KEEP REQUEUE DISCARD] 	V	N

4-16 Supported SQL, CICS, and IMS Statements
Supported RDMS 2200 SQL Statements

Statement	Format	Parser	CE
SELECT	<ul style="list-style-type: none"> Format SELECT [ALL DISTINCT] <select-list *> FROM table-specification-list [WHERE boolean-expression] [GROUP BY column-specification-list] [HAVING boolean-expression] INTO variable-specification-list 	V	N
SET	<ul style="list-style-type: none"> Format SET return-variable = [schema-name.]routine-name (parameter, ...) 	VO	N
SET STATISTICS	<ul style="list-style-type: none"> Format SET STATISTICS <ON OFF> 	VO	N
UNLOAD	<ul style="list-style-type: none"> Format UNLOAD TO FILE file-name <[FROM] CURSOR cursor-name query-expression [ORDER BY sort-specification-list]> [FORMAT <EXTERNAL [position-and-null-list] INTERNAL MAPPER>] [WITHOUT DESCRIPTION WITH DESCRIPTION [ONLY]] 	VO	N
UNLOCK	<ul style="list-style-type: none"> Format UNLOCK table-specification-list 	VO	N
UPDATE ALL	<ul style="list-style-type: none"> Format UPDATE table-specification [CHANGE] [IN] ALL [ROWS] change-specification-list [RETRIEVE INTO embedded-variable-specification-list] 	V	N
UPDATE Positioned	<ul style="list-style-type: none"> Format UPDATE table-specification [CHANGE] <SET COLUMNS> change-specification-list WHERE CURRENT OF cursor-name [RETRIEVE INTO embedded-variable-specification-list] 	V	N
UPDATE Searched	<ul style="list-style-type: none"> Format UPDATE table-specification [CHANGE] <SET COLUMNS> change-specification-list [WHERE boolean-expression] [RETRIEVE INTO embedded-variable-specification-list] 	V	N
UPDATE VALUES	<ul style="list-style-type: none"> Format UPDATE table-specification [CHANGE] VALUES (value-list) [WHERE boolean-expression] [RETRIEVE INTO embedded-variable-specification-list] 	V	N
USE DEFAULT	<ul style="list-style-type: none"> Format USE DEFAULT <<QUALIFIER SCHEMA> <qualifier-name placeholder> VERSION <version-name placeholder>> 	VO	N
WHENEVER	<ul style="list-style-type: none"> Format WHENEVER <SQLERROR NOT FOUND> <CONTINUE <GOTO GO TO> paragraph-name> 	V	N

4-18 Supported SQL, CICS, and IMS Statements
Supported RDMS 2200 SQL Statements

Supported CA-IDMS DML Statements



This chapter contains detailed information on CA-IDMS Data Manipulation Language statements supported by ATW for component extraction. For a list of supported legacy versions, refer to the *Release Notes*.

Supported CA-IDMS DML statements

Statement	Parser	CE
ABEND	V	S
ACCEPT	V	S
ACCEPT BIND ADDRESS	V	S
ACCEPT DATABASE STATISTICS	V	S
ACCEPT DB-KEY FROM CURRENCY	V	S
ACCEPT DB-KEY RELATIVE TO CURRENCY	V	S

5-2 Supported CA-IDMS DML Statements
Supported CA-IDMS DML statements

ACCEPT page-info-location	V	S
ACCEPT PROCEDURE CONTROL LOCATION	V	S
ACCEPT TRANSACTION STATISTICS	V	S
ATTACH	V	S
BIND MAP	V	S
BIND PROCEDURE	V	S
BIND RECORD	V	S
BIND RUN-UNIT	V	S
BIND TASK	V	S
BIND TRANSACTION STATISTICS	V	S
CHANGE PRIORITY	V	S
CHECK TERMINAL	V	S
COMMIT	V	S
CONNECT	V	S
DC RETURN	V	S
DELETE QUEUE	V	S
DELETE SCRATCH	V	S
DELETE TABLE	V	S
DEQUEUE	V	S
DISCONNECT	V	S
END LINE TERMINAL SESSION	V	S
END TRANSACTION STATISTICS	V	S
ENDPAGE	V	S
ENQUEUE	V	S
ERASE	V	S

ERASE (LRF)	V	S
FIND/OBTAIN	V	S
FIND/OBTAIN CALC/DUPLICATE	V	S
FIND/OBTAIN CURRENT	V	S
FIND/OBTAIN DB-KEY	V	S
FIND/OBTAIN OWNER	V	S
FIND/OBTAIN WITHIN SET USING SORT-KEY	V	S
FIND/OBTAIN WITHIN SET/AREA	V	S
FINISH	V	S
FREE STORAGE	V	S
GET	V	S
GET QUEUE	V	S
GET SCRATCH	V	S
GET STORAGE	V	S
GET TIME	V	S
IF	V	S
INQUIRE MAP	V	S
KEEP CURRENT	V	S
KEEP LONGTERM	V	S
LOAD TABLE	V	S
MAP IN	V	S
MAP OUT	V	S
MAP OUTIN	V	S
MODIFY	V	S
MODIFY (LRF)	V	S

5-4 Supported CA-IDMS DML Statements
Supported CA-IDMS DML statements

MODIFY MAP	V	S
OBTAIN (LRF)	V	S
POST	V	S
PUT QUEUE	V	S
PUT SCRATCH	V	S
READ LINE FROM TERMINAL	V	S
READ TERMINAL	V	S
READY	V	S
RETURN	V	S
ROLLBACK	V	S
SEND MESSAGE	V	S
SET ABEND EXIT	V	S
SET TIMER	V	S
SNAP	V	S
STARTPAGE	V	S
STORE	V	S
STORE (LRF)	V	S
TRANSFER CONTROL	V	S
WAIT	V	S
WRITE JOURNAL	V	S
WRITE LINE TO TERMINAL	V	S
WRITE LOG	V	S
WRITE PRINTER	V	S
WRITE TERMINAL	V	S
WRITE THEN READ TERMINAL	V	S

Bibliography

- *IBM Asset Transformation Workbench v1.1 Getting Started (SC31-6877-00)*
- *IBM Asset Transformation Workbench v1.1 Preparing Projects (SC31-6879-00)*
- *IBM Asset Transformation Workbench v1.1 Analyzing Projects (SC31-6880-00)*
- *IBM Asset Transformation Workbench v1.1 Analyzing Programs (SC31-6878-00)*
- *IBM Asset Transformation Workbench v1.1 Profiling Projects (SC31-6881-00)*
- *IBM Asset Transformation Workbench v1.1 Creating Components (SC31-6876-00)*
- *IBM Asset Transformation Workbench v1.1 Parser Reference (SC31-6882-00)*
- *IBM Asset Transformation Workbench v1.1 Architecture Reference (SC31-6898-00)*

Notices

This information was developed for products and services offered in the U.S.A. IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION •AS IS• WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
P.O. Box 12195, Dept. TL3B/B503/B313
3039 Cornwallis Rd.
Research Triangle Park, NC 27709-2195
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks of the IBM Corporation or its subsidiaries in the United States or other countries or both:

Table 1. Trademarks

IBM	MVS
AS400	CICS
IMS	DB/2
Database 2	OS/390
S/390	z/OS

The following terms are trademarks of other companies:

Java and JavaScript are registered trademarks and Sun Solaris and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.



Product Number: 5724-L54

SC31-6898-00



(1P) P/N:5724-L54

