

---

---

**Information technology — Programming  
languages, their environments and  
system software interfaces —  
Programming language Extended APL**

*Technologies de l'information — Langages de programmation, leurs  
environnements et interfaces logiciel système — Langage de  
programmation APL étendu*

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2001

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.ch](mailto:copyright@iso.ch)  
Web [www.iso.ch](http://www.iso.ch)

Printed in Switzerland

# Contents

<b>1</b>	<b>Scope</b>	<b>1</b>
<b>2</b>	<b>Normative References</b>	<b>2</b>
<b>3</b>	<b>Form of this International Standard</b>	<b>3</b>
3.1	Form of Definitions . . . . .	3
3.2	Named Arrays in Examples . . . . .	4
3.3	Notes . . . . .	5
3.4	Cross-References . . . . .	5
3.5	General Definitions . . . . .	5
<b>4</b>	<b>Compliance</b>	<b>7</b>
4.1	Conforming Implementations . . . . .	7
4.1.1	Required Behaviour for Conforming Implementations . . . . .	7
4.1.2	Required Documentation for Conforming Implementations . . . . .	8
4.1.2.1	Documentation of Optional-Facilities . . . . .	8
4.1.2.2	Documentation of Implementation-Defined-Facilities . . . . .	8
4.1.2.3	Consistent Extensions . . . . .	9
4.2	Conforming Programs . . . . .	9
4.2.1	Required Behaviour for Conforming Programs . . . . .	9
4.2.2	Required Documentation for Conforming Programs . . . . .	9
<b>5</b>	<b>Definitions</b>	<b>11</b>
5.1	Characters . . . . .	11
5.2	Numbers . . . . .	13
5.2.1	Elementary Operations . . . . .	13
5.2.2	Number Constants . . . . .	14
5.2.3	Subsets of the Set of Numbers . . . . .	14
5.2.4	Implementation Algorithms . . . . .	16
5.2.5	Defined Operations . . . . .	18
5.3	Objects . . . . .	20
5.3.1	Lists . . . . .	20
5.3.2	Arrays . . . . .	21
5.3.3	Defined-Functions . . . . .	23
5.3.4	Tokens . . . . .	25

5.3.4.1	Metaclasses	26
5.3.4.2	Index-List	29
5.3.5	Symbols	29
5.3.6	Contexts	29
5.3.7	Workspaces	30
5.3.8	Sessions	30
5.3.9	Shared-Variables	32
5.3.10	Systems	33
5.4	Evaluation Sequences	35
5.4.1	Evaluation Sequence Phrases	36
5.4.2	Diagrams	37
5.5	Other Terms	38
<b>6</b>	<b>Syntax and Evaluation</b>	<b>39</b>
6.1	Introduction	39
6.1.1	Evaluate-Line	39
6.1.2	Character-Diagrams	40
6.1.3	Evaluate-Statement	47
6.1.4	Bind-Token-Class	49
6.1.5	Literal-Conversion	50
6.1.6	Statement-Analysis Token-Diagrams	50
6.2	Reduce-Statement	55
6.3	The Phrase Evaluators	60
6.3.1	Diagrams	60
6.3.2	Remove-Parentheses	60
6.3.3	Evaluate-Niladic-Function	60
6.3.4	Evaluate-Monadic-Function	61
6.3.5	Evaluate-Monadic-Operator	62
6.3.6	Evaluate-Dyadic-Function	63
6.3.7	Evaluate-Dyadic-Operator	65
6.3.8	Evaluate-Indexed-Reference	66
6.3.9	Evaluate-Assignment	67
6.3.10	Evaluate-Indexed-Assignment	67
6.3.11	Evaluate-Variable	68
6.3.12	Build-Index-List	68
6.3.13	Process-End-of-Statement	69
6.4	The Form Table	70
<b>7</b>	<b>Scalar Functions</b>	<b>75</b>
7.1	Monadic Scalar Functions	76
7.1.1	Conjugate	76
7.1.2	Negative	76
7.1.3	Direction	77
7.1.4	Reciprocal	77
7.1.5	Floor	78
7.1.6	Ceiling	78
7.1.7	Exponential	79

7.1.8	Natural Logarithm . . . . .	79
7.1.9	Magnitude . . . . .	80
7.1.10	Factorial . . . . .	81
7.1.11	Pi times . . . . .	82
7.1.12	Not . . . . .	83
7.2	Dyadic Scalar Functions . . . . .	83
7.2.1	Plus . . . . .	84
7.2.2	Minus . . . . .	84
7.2.3	Times . . . . .	85
7.2.4	Divide . . . . .	85
7.2.5	Maximum . . . . .	86
7.2.6	Minimum . . . . .	86
7.2.7	Power . . . . .	87
7.2.8	Logarithm . . . . .	88
7.2.9	Residue . . . . .	89
7.2.10	Binomial . . . . .	90
7.2.11	Circular Functions . . . . .	91
7.2.12	And/LCM . . . . .	93
7.2.13	Or/GCD . . . . .	94
7.2.14	Nand . . . . .	94
7.2.15	Nor . . . . .	95
7.2.16	Equal . . . . .	96
7.2.17	Less than . . . . .	97
7.2.18	Less than or equal to . . . . .	98
7.2.19	Not equal . . . . .	99
7.2.20	Greater than or equal to . . . . .	100
7.2.21	Greater than . . . . .	101
<b>8</b>	<b>Structural Primitive Functions</b>	<b>102</b>
8.1	Introduction . . . . .	102
8.2	Monadic Structural Primitive Functions . . . . .	102
8.2.1	Ravel . . . . .	102
8.2.2	Shape . . . . .	103
8.2.3	Index Generator . . . . .	104
8.2.4	Table . . . . .	105
8.2.5	Depth . . . . .	106
8.2.6	Enlist . . . . .	107
8.3	Dyadic Structural Primitive Functions . . . . .	107
8.3.1	Reshape . . . . .	107
8.3.2	Join . . . . .	109
<b>9</b>	<b>Operators</b>	<b>110</b>
9.1	Introduction . . . . .	110
9.2	Monadic Operators . . . . .	110
9.2.1	Reduction . . . . .	110
9.2.2	Scan . . . . .	113
9.2.3	N-wise Reduction . . . . .	115

9.2.4	Duplicate . . . . .	118
9.2.5	Commute . . . . .	118
9.2.6	Each . . . . .	119
9.3	Dyadic Operators . . . . .	120
9.3.1	Outer Product . . . . .	120
9.3.2	Inner Product . . . . .	121
9.3.3	Rank operator definitions . . . . .	123
9.3.4	Rank operator deriving monadic function . . . . .	124
9.3.5	Rank operator deriving dyadic function . . . . .	125
<b>10</b>	<b>Mixed Functions</b>	<b>127</b>
10.1	Monadic Mixed Functions . . . . .	127
10.1.1	Roll . . . . .	127
10.1.2	Grade Up . . . . .	129
10.1.3	Grade Down . . . . .	131
10.1.4	Reverse . . . . .	132
10.1.5	Monadic Transpose . . . . .	133
10.1.6	Matrix Inverse . . . . .	134
10.1.7	Execute . . . . .	135
10.1.8	Unique . . . . .	136
10.1.9	First . . . . .	137
10.2	Dyadic Mixed Functions . . . . .	137
10.2.1	Join Along an Axis . . . . .	137
10.2.2	Index of . . . . .	140
10.2.3	Member of . . . . .	141
10.2.4	Deal . . . . .	142
10.2.5	Replicate . . . . .	143
10.2.6	Expand . . . . .	145
10.2.7	Rotate . . . . .	147
10.2.8	Base Value . . . . .	149
10.2.9	Representation . . . . .	151
10.2.10	Dyadic Transpose . . . . .	153
10.2.11	Take . . . . .	155
10.2.12	Drop . . . . .	156
10.2.13	Matrix Divide . . . . .	157
10.2.14	Indexed Reference . . . . .	158
10.2.15	Indexed Assignment . . . . .	159
10.2.16	Without . . . . .	161
10.2.17	Left . . . . .	161
10.2.18	Right . . . . .	162
10.2.19	Character Grade Definitions . . . . .	162
10.2.20	Character Grade Down . . . . .	163
10.2.21	Character Grade Up . . . . .	164
10.2.22	Pick . . . . .	166
10.2.23	Identical . . . . .	167
10.2.24	Disclose . . . . .	168
10.2.25	Disclose with Axis . . . . .	168

10.2.26	Enclose	169
10.2.27	Enclose with Axis	169
<b>11</b>	<b>System Functions</b>	<b>170</b>
11.1	Introduction	170
11.2	Definitions	170
11.3	Diagram	171
11.4	Niladic System Functions	171
11.4.1	Time Stamp	171
11.4.2	Atomic Vector	172
11.4.3	Line Counter	172
11.4.4	Event Message	173
11.4.5	Event Type	174
11.5	Monadic System Functions	174
11.5.1	Delay	174
11.5.2	Name Class	175
11.5.3	Expunge	176
11.5.4	Name List	177
11.5.5	Query Stop	178
11.5.6	Query Trace	179
11.5.7	Monadic Event Simulation	180
11.6	Dyadic System Functions	180
11.6.1	Name List	180
11.6.2	Set Stop	181
11.6.3	Set Trace	182
11.6.4	Execute Alternate	183
11.6.5	Dyadic Event Simulation	184
11.6.6	Transfer Form	185
<b>12</b>	<b>System Variables</b>	<b>186</b>
12.1	Definitions	186
12.2	Evaluation Sequences	187
12.2.1	Comparison Tolerance	187
12.2.2	Random Link	188
12.2.3	Print Precision	189
12.2.4	Index Origin	190
12.2.5	Latent Expression	191
<b>13</b>	<b>Defined Functions</b>	<b>192</b>
13.1	Introduction	192
13.2	Definitions	192
13.3	Diagrams	196
13.4	Operations	200
13.4.1	Call-Defined-Function	200
13.4.2	Defined-Function-Control	202
13.4.3	Function Fix	203
13.4.4	Character Representation	204
13.5	Function Editing	204

13.5.1	Evaluate-Function-Definition-Request . . . . .	204
13.5.2	Evaluate-Editing-Request . . . . .	206
13.5.3	Diagrams . . . . .	207
<b>14</b>	<b>Shared Variables</b>	<b>210</b>
14.1	Informal Introduction . . . . .	210
14.2	Definitions . . . . .	214
14.3	Diagrams . . . . .	214
14.4	Operations . . . . .	214
14.4.1	Primary-Name . . . . .	214
14.4.2	Surrogate-Name . . . . .	214
14.4.3	Degree-of-Coupling . . . . .	215
14.4.4	Access-Control-Vector . . . . .	215
14.4.5	Offer . . . . .	215
14.4.6	Retract . . . . .	216
14.4.7	Shared-Variable-Reset . . . . .	216
14.4.8	Report-State . . . . .	216
14.4.9	Signal-Event . . . . .	217
14.4.10	Clear-Event . . . . .	217
14.5	Shared Variable Forms . . . . .	217
14.5.1	Shared Variable Reference . . . . .	217
14.5.2	Shared Variable Assignment . . . . .	218
14.5.3	Shared Variable Indexed Assignment . . . . .	219
14.6	Shared Variable System Functions . . . . .	219
14.6.1	Shared Variable Access Control Inquiry . . . . .	219
14.6.2	Shared Variable Query . . . . .	221
14.6.3	Shared Variable Degree of Coupling . . . . .	222
14.6.4	Shared Variable Offer . . . . .	223
14.6.5	Shared Variable Retraction . . . . .	224
14.6.6	Shared Variable Access Control Set . . . . .	225
14.6.7	Shared Variable State Inquiry . . . . .	226
14.6.8	Shared Variable Event . . . . .	227
<b>15</b>	<b>Formatting and Numeric Conversion</b>	<b>228</b>
15.1	Introduction . . . . .	228
15.2	Numeric Conversion . . . . .	228
15.2.1	Numeric-Input-Conversion . . . . .	228
15.2.2	Numeric-Output-Conversion . . . . .	230
15.3	Diagrams . . . . .	231
15.4	Operations . . . . .	233
15.4.1	Monadic Format . . . . .	233
15.4.2	Dyadic Format . . . . .	237
<b>16</b>	<b>Input and Output</b>	<b>239</b>
16.1	Introduction . . . . .	239
16.2	Definitions . . . . .	240
16.2.1	User Facilities . . . . .	240
16.2.2	Implementation Algorithms . . . . .	240

16.2.3 Prompts . . . . .	241
16.3 Diagrams . . . . .	242
16.4 Operations . . . . .	242
16.4.1 Immediate-Execution . . . . .	242
16.4.2 Quad Input . . . . .	244
16.4.3 Quote Quad Input . . . . .	245
16.4.4 Quad Output . . . . .	245
16.4.5 Quote Quad Output . . . . .	246
<b>17 System Commands</b>	<b>247</b>
17.1 Introduction . . . . .	247
17.2 Definitions . . . . .	247
17.3 Diagrams . . . . .	248
17.4 Operations . . . . .	248
17.4.1 Evaluate-System-Command . . . . .	248
17.5 Diagrams and Evaluation Sequences . . . . .	249
<b>Annex A (normative) Component Files</b>	<b>254</b>
A.1 Definitions of arguments and results . . . . .	254
A.2 Definition of functions . . . . .	255
A.3 Errors . . . . .	256

## List of Figures

1	Statement Evaluation. . . . .	59
2	Shared Variable Access Rules. . . . .	213

## List of Tables

1	The Required Character Set . . . . .	12
2	Relationship between Class-Name and Content . . . . .	27
3	The Phrase Table. . . . .	58
4	The Form Table . . . . .	71
5	Actions for the Reduction of an Empty Vector. . . . .	113
6	Actions for the N-wise Reduction of an Empty Vector. . . . .	117

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 13751 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Annex A forms a normative part of this International Standard.

## Introduction

APL stands for **A Programming Language**. It is a notation invented by K. E. Iverson in the late 1950s for the description of algorithms, and expanded on and made into the programming system *APL\360* by Iverson and his colleagues Adin Falkoff, Larry Breed, Dick Lathwell, and Roger Moore in the mid-1960s.

This document, **Programming Language APL, Extended**, is a sequel to **Programming Language APL**, ISO 8485 (1989).

The principal differences that the reader will find here have to do with new features that have been added. These topics are:

- without
- greatest common divisor
- least common multiple
- duplicate
- commute
- table
- join along first axis
- mixed arrays
- overbar in names
- underbar in names
- replicate
- character grades
- grades of arrays greater than rank one
- unique
- alpha as a name
- omega as a name
- ambivalent defined functions
- event handling
- n-wise reduction
- complex arithmetic
- left
- right
- function rank operator
- defined operators
- component file system

enclose  
disclose  
enlist  
pick  
depth  
identical  
each  
first

An entry for each of these topics will be found in the index. Some new system commands have been added. Shared variable extensions have been added. Workspace Interchange Standard 2 is given, in which canonical representation vectors of type “E” are used to represent generalised arrays.

# **Information technology — Programming languages, their environments and system software interfaces — Programming language Extended APL**

## **1 Scope**

This International Standard defines the programming language APL and the environment in which APL programs are executed. Its purpose is to facilitate interchange and promote portability of APL programs and programming skills. This International Standard specifies the syntax and semantics of APL programs and the characteristics of the environment in which APL programs are executed.

It also specifies requirements for conformance to this International Standard, including the publication of values and characteristics of implementation properties so that conforming implementations can be meaningfully compared.

This International Standard does not specify:

- implementation properties that are likely to vary with the particular equipment or operating system used;
- required values for implementation limits such as APL workspace size or numeric precision;
- the data structures used to represent APL objects;
- the facilities available through shared variables.

## 2 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 2382-15:1999, *Information technology – Vocabulary – Part 15: Programming languages*.

ISO 8485:1989, *Programming languages – APL*.

*International Register of Coded Character Sets To Be Used With Escape Sequences, Registered character set 68.* (<http://www.itscj.ipsj.or.jp/ISO-IR/068.pdf>)