
**Information technology — Computer
graphics and image processing — Image
processing and interchange — Application
program interface language bindings —**

Part 4:

C

*Technologies de l'information — Infographie et traitement de l'image —
Traitement et échange de l'image — Liants de langage d'interface de
programme d'application —*

Partie 4: C

Contents

1 Scope	1
2 Normative references	2
3 The C language binding of the Image Processing and Interchange Standard	3
3.1 Classification and designation	3
3.2 Functions versus macros	3
3.3 Implications of the language	4
3.3.1 Character strings	4
3.3.2 Implementation dependencies	4
3.3.3 Data object repository	4
3.4 Identifier mapping	5
3.5 Return values	6
3.6 Header files	6
3.7 Memory management	7
3.8 Error handling	9
3.8.1 Application defined error handlers	9
3.8.2 Function identification	9
3.8.3 Error presentation	9
3.9 Virtual register support	10
3.10 Convenience functions	10
3.11 Program examples	10
4 Tables	11
4.1 Abbreviations	11
4.1.1 Table of abbreviations	11
4.1.2 Abbreviation policy in construction of identifiers	14
4.2 Function names	15
4.2.1 Alphabetical by function name	15
4.2.2 Alphabetical by bound name	27
5 Data type definitions	39
5.1 Mapping of data types	39
5.2 Environment data type definitions	40
5.2.1 External physical image data types	40
5.2.2 Basic parameter data types	41
5.2.2.1 IPI-PIKS parameter data types	41
5.2.2.2 IPI-IIF parameter data types	41
5.2.3 Data object identifiers	42
5.2.3.1 IPI-PIKS object identifiers	42
5.2.3.2 IPI-IIF object identification	43
5.3 Implementation dependent data type definitions	44
5.4 Implementation independent data type definitions	45
5.4.1 Enumerated data type definitions	45
5.4.1.1 IPI-PIKS enumerated data type definitions	45
5.4.1.2 IPI-IIF enumerated data type definitions	50

5.4.2 Union data type definitions.....	51
5.4.2.1 IPI-PIKS union data type definitions.....	51
5.4.2.2 IPI-IIF union type data definitions.....	66
5.4.3 Structure data type definitions.....	68
5.4.3.1 IPI-PIKS structures.....	68
5.4.3.2 IPI-IIF structures.....	83
6 Macro definitions.....	87
6.1 Unbounded option and status codes.....	87
6.1.1 IPI-PIKS unbounded option and status codes.....	101
6.1.2 IPI-IIF unbounded option and status codes.....	103
6.2 IPI-IIF syntax descriptors.....	103
6.2.1 IPI-IIF syntax entity types.....	107
6.2.2 IPI-IIF syntax components.....	123
6.3 Function descriptors.....	123
6.3.1 IPI-PIKS function descriptors.....	135
6.3.2 IPI-PIKS convenience function descriptors.....	136
6.3.3 IPI-IIF function descriptors.....	137
6.3.4 IPI-IIF convenience function descriptors.....	138
6.4 Error descriptors.....	138
6.4.1 IPI-PIKS errors.....	143
6.4.2 IPI-IIF errors.....	145
6.4.3 System errors.....	145
6.4.4 Binding-specific errors.....	146
6.5 Implementation-specific data lengths.....	147
6.6 Repository object descriptors.....	147
6.6.1 Repository impulse response function arrays.....	149
6.6.2 Repository dither arrays.....	149
6.6.3 Repository colour conversion matrices.....	155
7 C functional elements.....	155
7.1 Notational conventions.....	156
7.2 IPI-PIKS functional element prototypes.....	273
7.3 IPI-IIF functional element prototypes.....	282
8 Convenience functions.....	282
8.1 IPI-PIKS convenience functions.....	282
8.1.1 Image preparation functions.....	283
8.1.1.1 prepare_colour_image.....	284
8.1.1.2 prepare_monochrome_image.....	285
8.1.1.3 create_unbounded_image_copy.....	287
8.1.2 ROI creation functions.....	288
8.1.2.1 generate_2d_roi_rectangular.....	289
8.1.2.2 generate_roi_coordinate.....	290
8.1.2.3 generate_roi_elliptical.....	291
8.1.2.4 generate_roi_polygon.....	293
8.1.2.5 generate_roi_rectangular.....	294
8.1.2.6 prepare_2d_roi_rectangular.....	295
8.1.2.7 prepare_roi.....	296
8.1.3 Tuple generation functions.....	297
8.1.3.1 generate_nd_1_tuple.....	298
8.1.3.2 generate_nd_3_tuple.....	299
8.1.3.3 generate_nd_4_tuple.....	300
8.1.3.4 generate_nd_5_tuple.....	301
8.1.3.5 generate_rd_3_tuple.....	302
8.1.3.6 generate_rd_4_tuple.....	303
8.1.3.7 generate_rd_5_tuple.....	303

Contents

8.1.3.8 generate_sd_1_tuple.....	304
8.1.3.9 generate_sd_3_tuple.....	305
8.1.3.10 generate_sd_4_tuple.....	306
8.1.3.11 generate_sd_5_tuple.....	307
8.1.4 Tiled image import and export.....	308
8.1.4.1 tiled_image_export.....	308
8.1.4.2 tiled_image_import.....	309
8.2 IPI-IIF convenience functions.....	310
8.2.1 Attach entity functions.....	310
8.2.1.1 attach.....	310
8.2.1.2 attach_sequence.....	310
8.2.1.3 attach_sequence_end.....	311
8.2.2 Create identifier function.....	311
8.2.3 Get entity component functions.....	312
8.2.3.1 get_component.....	312
8.2.3.2 get_sequence_component.....	312
8.2.4 Get entity value functions.....	313
8.2.4.1 get_entity_boolean.....	313
8.2.4.2 get_entity_integer.....	314
8.2.4.3 get_entity_real.....	314
8.2.4.4 get_entity_string.....	314
8.2.4.5 get_sequence_boolean.....	314
8.2.4.6 get_sequence_integer.....	315
8.2.4.7 get_sequence_real.....	315
8.2.4.8 get_sequence_string.....	315
8.2.4.9 get_sequence_end.....	316
8.2.5 Put entity value functions.....	317
8.2.5.1 put_entity_boolean.....	318
8.2.5.2 put_entity_integer.....	318
8.2.5.3 put_entity_real.....	318
8.2.5.4 put_entity_string.....	319
8.2.5.5 put_sequence_boolean.....	319
8.2.5.6 put_sequence_integer.....	319
8.2.5.7 put_sequence_real.....	320
8.2.5.8 put_sequence_string.....	320
Annexes	
A Memory management.....	321
A.1 Introduction.....	321
A.2 Functions that import the application data to the implementation memory.....	323
A.3 Functions that export the implementation data to the application memory.....	325
B Macros, data types in compilation order and external functions.....	327
B.1 IPI-PIKS macros, types and functions.....	327
B.1.1 Macro definitions.....	327
B.1.2 Data types in compilation order.....	367
B.1.3 Functions.....	390
B.1.4 Convenience functions.....	475
B.2 IPI-IIF macros, types and functions.....	480
B.2.1 Macro definitions.....	480
B.2.2 Data types in compilation order.....	499
B.2.3 Functions.....	504
B.2.4 Convenience functions.....	510

C Sample programs.....	513
C.1 IPI-PIKS application examples.....	513
C.1.1 Application use of IPI-PIKS for memory-managed image import/export	514
C.1.2 Histogram generation.....	519
C.1.3 Region of interest control.....	523
C.1.4 Simulated unsharp mask operation.....	526
C.1.5 Demonstration of asynchronous and chained application.....	531
C.1.5.1 Image blend by synchronous, unchained functional representation.....	531
C.1.5.2 Image blend by asynchronous functional representation.....	533
C.1.5.3 Image blend by chained functional representation.....	537
C.2 IPI-IIF application example.....	541
D Macros for short function identifiers.....	550
Index.....	562

List of figures

1 - Buffered memory management of InImportImage(...)	324
2 - Buffered memory management of InExportImage(...)	326

List of tables

1 Data type prefixes	5
2 Abbreviations	11
3 Function names alphabetical by function name	15
4 Function names alphabetical by bound name	27
5 Data types	39
6 IPI-PIKS function descriptors	123
7 IPI-PIKS convenience function descriptors	135
8 IPI-IIF function descriptors	136
9 IPI-IIF convenience function descriptors	137
10 IPI-PIKS error descriptors	138
11 IPI-IIF error descriptors	143
12 System error descriptors	145
13 Binding-specific error descriptors	145
14 IPI-PIKS data lengths	146
15 Repository impulse response function array descriptors	147
16 Repository dither array indices	149
17 Repository colour conversion matrix indices	149

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.

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.

International Standard ISO/IEC 12088-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 24, *Computer graphics and image processing*.

ISO/IEC 12088 consists of the following part, under the general title *Information technology — Computer graphics and image processing — Image processing and interchange — Application program interface language bindings*:

— Part 4: C

Other parts may follow.

Annexes A to D of this part of ISO/IEC 12088 are for information only.

Introduction

The Image Processing and Interchange (IPI) functional specification, ISO/IEC 12087, upon which this binding is based, emerged as an International Standard in 1994. It consists of three parts: Part 1: Common Architecture for Imaging (IPI-CAI), Part 2: Programmer's Imaging Kernel System (IPI-PIKS) Application Program Interface and Part 3: Image Interchange Facility (IPI-IIF).

The functional description of ISO/IEC 12088 is specified in a language independent manner and needs to be embedded in language dependent layers (language bindings) for use with particular programming languages.

The purpose of this part of ISO/IEC 12088 is to define a standard binding for the Image Processing and Interchange Standard in the C programming language.

Information technology — Computer graphics and image processing — Image processing and interchange — Application program interface language bindings —

Part 4:

C

1 Scope

ISO/IEC 12087 consists of the three parts which define the functional aspects of this part of ISO/IEC 12088. The Common Architecture for Imaging (IPI-CAI) defines the overall architecture. The Programmer's Imaging Kernel System (IPI-PIKS) and the Image Interchange Facility (IPI-IIF) each specify a language independent, image processing Application Program Interface (API) within the Image Processing and Interchange Standard. Either API may be implemented independently or both may be combined in one implementation. For integration into a programming language, IPI-PIKS and IPI-IIF APIs are embedded in a language dependent layer obeying the particular conventions of that language. This part of ISO/IEC 12088 specifies such a language dependent layer for the C language.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 12088. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 12088 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

[1] ISO/IEC 12087-1:1995, *Information technology - Computer graphics and image processing - Image Processing and Interchange (IPI) - Functional specification - Part 1: Common architecture for imaging*.

[2] ISO/IEC 12087-2:1994, *Information technology - Computer graphics and image processing - Image Processing and Interchange (IPI) - Functional specification - Part 2: Programmer's imaging kernel system application programme interface*.

[3] ISO/IEC 12087-3:1995, *Information technology - Computer graphics and image processing - Image Processing and Interchange (IPI) - Functional specification - Part 3: Imaging Interchange Facility (IIF)*.

[4] ISO/IEC 9899:1990, *Programming Languages - C*.