

INTERNATIONAL STANDARD

ISO/IEC 12088-4

First edition
1995-12-15

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



Reference number
ISO/IEC 12088-4:1995(E)

AVA Documents
Document Sharing Hub

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

© ISO/IEC 1995

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 the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

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

3 The C language binding of the Image Processing and Interchange Standard

3.1 Classification and designation

This part of ISO/IEC 12088 incorporates the rules of conformance defined in the Image Processing and Interchange Standard (ISO/IEC 12087) for implementations, with those additional requirements specifically defined for C language implementations of the Standard.

The following criteria shall determine conformance of an implementation of this part of ISO/IEC 12088:

In order to conform, development of a C language implementation of the Image Processing and Interchange Standard shall implement one or more specific profiles as specified in the functional specification of ISO/IEC 12087. Independent implementation of the Programmer's Imaging Kernel System (Part 2 of ISO/IEC 12087 including relevant portions of Part 1) or the Image Interchange Facility (Part 3 of ISO/IEC 12087 including relevant portions of Part 1) is allowed.

The implementation shall make visible all of the declarations in the C binding specified in this part of ISO/IEC 12088 for that same level of ISO/IEC 12087 and all lower levels and for a specific level of the C language. Thus, for example, the syntax of the function names shall be precisely as specified in the binding and parameters shall be of the data types stated in the binding.

This part of ISO/IEC 12088 supports the conformance profiles defined in the functional specification of ISO/IEC 12087.

3.2 Functions versus macros

An implementation may substitute macros for functions. Implementation by macro should result in the same effect as implementation by function including any side effects.

3.3 Implications of the language

3.3.1 Character strings

The C language represents character strings as an array of characters terminated by the null character (i.e., '\0'). Therefore, the null character is not usable as a printable character.

3.3.2 Implementation dependencies

There are a number of implementation issues which will affect the portability of application programs employing this part of ISO/IEC 12087. The application programmer should follow accepted practices for ensuring the portability of C language programs to avoid introducing problems when recompiling the application on another system.

This part of ISO/IEC 12088 attempts to avoid dependencies on computer specific types which may vary among computers. Clause 5 recommends type mappings for common numeric types, data object identifiers, and precision to be used throughout this C language binding.

3.3.3 Data object repository

The Image Processing and Interchange Standard provides a repository of data objects for routinely used objects. These objects are defined in ISO/IEC 12087. Access to these objects for their assignment to IPI-PIKS operators is by IPI-PIKS-internal identifiers which may be acquired by the execution of a *return repository id* function presented in 7.2 of this part of ISO/IEC 12088.

The repository object identifiers are defined in 6.5. Each repository object name begins with the characters "IR_".

3.4 Identifier mapping

The IPI-PIKS-API and IPI-IIF-API function names are mapped to C identifiers beginning with the sentinel "I" followed by a lower case name prefix which identifies the return parameter type. Words and phrases used in the function names are concatenated with and beyond the prefix and may be abbreviated in the name representation. The acronyms and the first letter of name words and abbreviations are in upper case and the remainder are in lower case. For example, the identifier of the IPI-PIKS *import image* function, which returns an IPI-PIKS internal object identifier, is *ImportImage*.

Data type element names are mapped to C identifiers beginning with the sentinel "I." The sentinel is followed by a two-part prefix, which delineates the usage and type of the element, followed by a name. These usage and data type designators, see Table 1, include "m" for external physical image pixel representation, "p" for parameter representation and "d" for internal representation. For some data types, the type designator and name may be merged, e.g., *Ipint*, the integer basic type. Acronyms, words and phrases used in the type names are in lower case, often abbreviated in the representation, and separated with the underscore character, "_". For example, the union of pixel data types used for parameters is *Ipsparameter pixel*.

Enumerated data type constants and macro names are mapped to C identifiers beginning with the sentinel "I" followed by a word or abbreviation of the entity to which the identifier belongs and the name of the identifier in upper case with acronyms, abbreviations or words separated with the underscore character, "_". For example the macro name for Boolean false is *IBOOL FALSE*.

Parameters are mapped to C identifiers beginning with a prefix which identifies the type and/or usage. Words and phrases used in the parameter and variable names are concatenated with and beyond the prefix and may be abbreviated in the representation. The first letter of name words and abbreviations are in upper case and the remainder are in lower case. Acronyms are presented in upper case. For example, the name of the function parameter which denotes the identifier of a destination ROI object is *nDestROI*.

The prefix symbols used in this part of ISO/IEC 12088 are listed in Table 1.

Table 1 - Data type prefixes

Prefix	Meaning
a	array
b	Boolean
c	character
d	internal data type
e	enumerated data type
f	function
i	integer
m	external image data type
n	identifier
p	parameter type
r	real
s	structure
t	pointer
u	unsigned integer
v	void
x	complex
z	zero terminated string

The C language binding of the Image Processing and Interchange Standard

The C language standard requires that compilers recognize internal identifiers which are distinct in at least 31 characters. That standard also requires that external identifiers (i.e., those seen by the linker) be recognized to a minimum of six characters, independent of case. Implementations which must run in environments which accept less than 31 characters in external identifiers must include a set of macros (`#defines`) in the header file which equate the long names as defined in this document to a set of shorter names which conform to the requirements of the implementation. An example of these shorter 6-character names can be found in Annex D

3.5 Return values

Functional elements which perform a status test within the IPI-PIKS internal domain, and do not affect destination data, output the Boolean result of the test as return value.

Destination identifiers or values are output as return values from elements which include destination objects or values, respectively, as part of their parameter list, even if the identifier itself is input to the element. If more than one destination object is identified by a function, no return value is output. Return values are void from elements which do not provide destination objects or values.

If the destination identifier, output as a return value, is generated by the functional element, the parameter represented by the return value is not included in the argument list of the functional element. Conversely, if the same parameter appears in the argument list, the return value is a replication of an input of the destination identifier, though the contents of the object may be modified by the function.

3.6 Header files

The C language provides a mechanism to allow external files to be included in a compilation. Clause 5 of this part of ISO/IEC 12088 describes the data types that shall be defined in the `piks.h` and `iif.h` which should be included in any application program that intends to use IPI-PIKS and/or IPI-IIF, respectively, via the C language binding. Additional implementation-dependent items may be placed in these files if needed and the files may be concatenated as one header file. These items should start with the sentinel "I", as far as applicable.

The files `piks.h` and `iif.h` shall also contain external prototypes for all IPI-PIKS and IPI-IIF C language functions. For example, the prototype for the IPI-PIKS function *InResize* would appear as follows:

```
extern Idnimage InResize (                /* OUT destination image identifier */
    Idnimage      nSourceImage,          /* source image identifier           */
    Idnimage      nDestImage,            /* destination image identifier      */
    Idntuple      nDestSize              /* destination size ND 5-tuple identifier */
);
```

Examples of `piks.h` and `iif.h` which utilize suggested data types are presented in Annex B.

3.7 Memory management

IPI-PIKS requires internal memory space, denoted by identifiers containing the letter "d" in their prefixes (immediately following the sentinel) as they appear in this part of ISO/IEC 12088. Internal memory requirements may exist in a memory domain independent of application memory or may be acquired from application memory as provided in a specific implementation. ISO/IEC 12087 and this part of ISO/IEC 12088 do not provide functionality to support the allocation and management of IPI-PIKS internal memory should it be acquired from the application domain.

Both IPI-IIF and IPI-PIKS require external memory space, denoted by identifiers containing the letter "p" in their prefixes (immediately following the sentinel) as they appear in this part of ISO/IEC 12088. External memory requirements must be acquired from application memory space. The application shall allocate the external memory for the data provided to and returned by the implementation of the Image Processing and Interchange Standard. The application will allocate C structures and pass the appropriate identifiers of those structures to the requiring functional element, which will then place information into or receive information from the structures. However, the application may provide data buffered in partial amounts for large data objects input to the functions. Also, some functions return variable length data, the length of which is not known a priori and can not be computed by the application.

If the memory space requirement is small, even if variable in length, the structure and sufficient space must be provided by the application. As example, a B-tuple for colour image data precision specification will contain no more than four data elements and, as such, is a small structure. A B-tuple for spectral image data precision specification may contain many elements.

Functions which may receive buffered data and functions which return variable lengths of sizable data or are considered to return potentially large amounts of fixed-length data are managed such that the application can submit or receive buffered portions of the data. The data portion, or buffer, is a subset of the implementation's internal data structure and is called the application's buffer. This allows the application to process the implementation's data, considered as a single dimension array, in a piecewise manner rather than all at once.

The application allocates the memory for its buffer and passes an identifier of that buffer to the implementation. The implementation places the subset of data into the buffer according to the specifications included among the function's parameters. In order to support this policy of memory management, three additional parameters have been added to functions which may transfer buffered data:

- a) **iApplBufferSize**: An integer input parameter which specifies the length of the application's buffer, dimensioned in terms of the number of data elements of data type specified for the input or return. The positive value of **iApplBufferSize** indicates the element count of the data which can be contained in the application's buffer. A value of 0 is valid as input for this parameter and is useful to interrogate the overall data size available for output or available in the implementation for input of each buffered transfer operation. A negative value is valid for this parameter and is interpreted as a signal for complete transfer of the data, the application being responsible to provide adequate space for the complete data transfer from implementation to application or the complete data requirement for transfer from application to implementation.
- b) **uStartPoint**: An unsigned integer input parameter which is an element index offset into the implementation's data space represented as a 1D array of data type specified for the external data being transferred. **uStartPoint** of 0 represents the first element of both the implementation's and application's data. Otherwise, **uStartPoint** is the index which indicates the element offset in the implementation's data space where data transfer is to be initiated to or from the beginning of the application's buffer. The transfer of data continues to the extent of the application's buffer or until there are no more items to be transferred, whichever is least.

The C language binding of the Image Processing and Interchange Standard

- c) **tuDataSize:** An output parameter which is a pointer to an unsigned integer value. As directed by this indirection parameter, the implementation transfers a value which is the size of the data remaining beyond the given **uStartPoint** less the amount of data transferred during execution of the function. The size is specified in terms of elements of the external data type specified for the data to be transferred into or out of the implementation's data space.

Errors that can be invoked during execution of the functions which utilize this memory management policy are listed in 6.4.4. Failures to correctly transfer data as specified are reported as appropriate error designators by all using functions except inquiry functions. Inquiry functions do not enter into any error condition but provide parametric return to indicate the validity of the data transfer.

A more detailed description of memory management is provided in Annex A and sample programming is presented in Annex C.

3.8 Error handling

The error handling mechanism specified for IPI APIs employs methodology similar to the error handling specified in the graphics standards. This mechanism is described in Parts 2 and 3 of ISO/IEC 12087.

3.8.1 Application defined error handlers

An application can define an IPI binding-specific error handling function for the implementation and access this via the following utility function which has been added to the functions of IPI-PIKS and IPI-IIF in this part of ISO/IEC 12088:

```
void *ItfSetErrorHandler (                                /* OUT old error handling function pointer */  
    void (*ItfNewErrorHandler)() /* new error handler function pointer */  
);
```

Application-defined error handling functions shall accept the same arguments as the standard error handler. In IPI-PIKS, this function is not chainable and requires synchronous operation. It shall be provided in all implementation profiles.

Initially, the implementation will invoke its standard error handler whenever an error occurs. After a call of *ItfSetErrorHandler*, the implementation will invoke the user-defined error function, *ItfNewErrHandler*, whenever an error is detected. The pointer to the previous error handler is returned to the application for later use.

If *ItfNewErrHandler* is not a valid pointer, the error handling will automatically be done by the standard error handler.

3.8.2 Function identification

This part of ISO/IEC 12088 defines descriptors for the identification of functional elements in 6.3. These constant values are the same as those specified in Parts 2 and 3 of ISO/IEC 12087 for functions which appear in those documents.

Function identifier values defined in ISO/IEC 12087 are assigned herein except where functions have been added in this part of ISO/IEC 12088 or have been subdivided to include versions which permit virtual register usage for return value capture and parameter specification. These additional functions are assigned identifier values sequentially following the function identifier values given in ISO/IEC 12087.

Each function identification descriptor begins with the characters "IF_" or "IC_", the latter for convenience functions.

3.8.3 Error presentation

This part of ISO/IEC 12088 defines descriptors for the error conditions in 6.4. Error descriptor values defined in ISO/IEC 12087 are assigned herein.

Each error descriptor begins with the characters "IE_".

3.9 Virtual register support

Virtual registers can be used to replace the parameters and receive the return values of functions. This is necessary for parameter alteration when these functions are operating in chains. To support the substitution of virtual registers as function arguments and recipients of return values, this part of ISO/IEC 12088 adds a separate version of each chainable function, named with a suffix of VReg.

3.10 Convenience functions

Specification of common operations which require the execution of several functions bound in this part of ISO/IEC 12088 are grouped into convenience functions for application users who desire to reduce the amount of programming from such details. A number of these functions are presented in clause 8 where they are normative to assure portability.

3.11 Program examples

Example programs of IPI application are presented in Annex C. These illustrate some of the common operations and uses of the fundamental functionality bound in this part of ISO/IEC 12088.

4 Tables

4.1 Abbreviations

4.1.1 Table of abbreviations

Only words and phrases which are abbreviated appear in this table.

The abbreviations may be used for

comments (acronyms only),
data types,
enumerated constants,
function names,
macro names,
parameter names,
text.

Table 2 - Abbreviations

Abbreviation	Word or Phrase
1D	one dimension
2D	two dimension
2nd	second
3D	three dimension
4D	four dimension
5D	five dimension
A	appearance
API	Application Program Interface
appl	application
arg	argument
arith	arithmetic
ASN.1	Abstract Syntax Notation One
asynch	asynchronous
attr	attribute
B	image spectral band size
BD	Boolean, internal
BI	Boolean, pixel type
bool	Boolean
BP	Boolean, parameter
C	colourmetric
CAI	Common Architecture for Imaging
CCIR	Comite Consultatif International des Radiocommunications
ccw	counterclockwise
CD	complex, internal
CF	complex, external
char	character
chc	choice
chr	chrominance
chrom	chromaticity
CIE	Commision Internationale de l'Eclairage

Table 2 - Abbreviations (continued)

Abbreviation	Word or Phrase
cmp	component
CMY	cyan, magenta, yellow colour coordinates
CMYK	cyan, magenta, yellow, black colour coordinates
COLR	colour image structure
conv	conversion
CP	complex, parameter
CS	character string
def	defined, definition
deriv	derivative
dest	destination
detect	detection
diff	difference, differencing
dilate	dilation
DOR	designated object reference
EBCU	European Broadcasting Union
enum	enumerated
equal	equalization
fund	fundamental
Gauss	Gaussian
GEN	generic image structure
grad	gradient
hist	histogram
id	identifier
ID	internal identifier
ID_ARRAY	composite image identifier array
ID_LIST	composite image identifier list
ID_RECORD	composite image identifier record
IEC	International Electrotechnical Commission
IIF	Image Interchange Facility
impl	implementation
int	integer
interp	interpolated, interpolation
IHS	intensity, hue, saturation colour coordinates
IP	parameter identifier
IPI	Image Processing and Interchange
ISO	International Organization for Standardization
JBIG	Joint Bi-level Image Experts Group
JPEG	Joint Photographic Experts Group
LAB	L*a*b* colour coordinates
Laplace	Laplacian
lin	linear
lowleft	lower left
lowright	lower right
lsb	least significant bit
lum	luminance
LUT	lookup table
LUV	L*u*v* colour coordinates
manip	manipulation

Table 2 - Abbreviations (continued)

Abbreviation	Word or Phrase
MON	monochrome image structure
MPEG	Moving Pictures Experts Group
msb	most significant bit
nbhood	neighbourhood
nbr	neighbour
ND	unsigned integer, internal
NI	unsigned integer, pixel type
nonlin	nonlinear
NP	unsigned integer, parameter
NTSC	National Television Systems Committee
orth	orthogonal
OUT	output parameter or structure
PIKS	Programmer's Imaging Kernel System
pixel	picture element
RD	real, internal
RF	real floating point, external
RGB	red, green, blue colour coordinates
ROI	region of interest
RP	real, parameter
SD	signed integer, internal
sdev	standard deviation
sep	separated
seq	sequence
SI	signed integer, pixel type
SMPTE	Society of Motion Picture and Television Engineers
SP	signed integer, parameter
SPE	spectral image structure
stat	statistical
synch	synchronous
T	image temporal size
TC	temporal colour structure
TEM	temporal image structure
templ	template
TI	fixed point integer, pixel type
trunc	truncated
TS	temporal spectral structure
uint	unsigned integer
unbnd	unbounded
upleft	upper left
upright	upper right
UVW	UVW colour coordinates
VC	volumetric colour image
VOL	volumetric image structure
VReg	virtual register
VS	volumetric-spectral image structure
VT	volumetric-temporal image structure
VTC	volumetric-temporal colour image structure
VTS	volumetric-temporal-spectral image structure

Table 2 - Abbreviations (concluded)

Abbreviation	Word or Phrase
X	image horizontal size
XYZ	XYZ colour coordinates
Y	image vertical size
YCBCR	YCbCr colour coordinates
YIQ	YIQ colour coordinates
YUV	Yuv colour coordinates
YXY	Yxy colour coordinates
Z	image depth size

4.1.2 Abbreviation policy in construction of identifiers

In the construction of the names for data types, structures, macros, functions, arguments, etc., the following policy is applied:

- 1) Abbreviations and acronyms are defined and applied only when needed to avoid name lengths in excess of 31 characters.
- 2) When a word is abbreviated or a compound term is converted to an acronym, only one abbreviation applies for that word or one acronym for that term and all names which would utilize the word or term, contain only the abbreviation or acronym.
- 3) The plural of an expression is constructed by adding an "s" after the abbreviation: so, for example, "attribute" is abbreviated to "attr" and "attributes" is abbreviated to "attrs". If an expression is mapped to NULL, so will be its plural.

4.2 Function names

Function names presented in ISO/IEC 12087 are bound in this part of ISO/IEC 12088 according to construction rules presented in 3.4. Function names are related to bound names in the tables of the following clauses. Functions of IPI-PIKS and IPI-IIF are both included in tables 3 and 4.

The existence of functions within an implementation is directed by a profile level specified for that implementation in ISO/IEC 12087. IPI-PIKS defines four profiles for differing levels of implementation which, in increasing order of capability, are foundation, technical, scientific and full. In addition, the inclusion of the IPI-PIKS *input object* and *output object* functions is required only if the IPI-PIKS implementation is IPI-IIF capable. Functions appearing in a lower profile must be included in subsequent levels.

IPI-IIF defines three profiles for differing levels of implementation; however, all IPI-IIF functions are required in all IPI-IIF profiles. The inclusion of *export iif data to piks* and *import iif data from piks* functions is required only if the IPI-IIF implementation is IPI-PIKS capable.

Tables 3 and 4 contain the identification of the least profile level in which the listed function is required

4.2.1 Alphabetical by function name

Table 3 - Function names alphabetical by function name

Function Name	Bound Name	Least Profile
abort_asynchronous_execution	Iv.Abort.AsynchExecution	Scientific
accumulator	Itra.Accumulator	Foundation
adaptive_histogram_equalization	InAdaptiveHistEqual	Scientific
adaptive_histogram_equalization_vreg	InAdaptiveHistEqualVReg	Full
allocate_chain	InAllocateChain	Full
allocate_chain_vreg	InAllocateChainVReg	Full
allocate_composite_identifier_array	InAllocateIdArray	Scientific
allocate_composite_identifier_array_vreg	InAllocateIdArrayVReg	Full
allocate_composite_identifier_list	InAllocateIdList	Scientific
allocate_composite_identifier_list_vreg	InAllocateIdListVReg	Full
allocate_composite_identifier_record	InAllocateIdRecord	Scientific
allocate_composite_identifier_record_vreg	InAllocateIdRecordVReg	Full
allocate_histogram	InAllocateHist	Foundation
allocate_histogram_vreg	InAllocateHistVReg	Full
allocate_image	InAllocateImage	Foundation
allocate_image_vreg	InAllocateImageVReg	Full
allocate_lookup_table	InAllocateLUT	Foundation
allocate_lookup_table_vreg	InAllocateLUTVReg	Full
allocate_matrix	InAllocateMatrix	Foundation
allocate_matrix_vreg	InAllocateMatrixVReg	Full
allocate_neighbourhood_array	InAllocateNbhodArray	Foundation
allocate_neighbourhood_array_vreg	InAllocateNbhodArrayVReg	Full
allocate_pixel_record	InAllocatePixelRecord	Technical
allocate_pixel_record_vreg	InAllocatePixelRecordVReg	Full
allocate_roi	InAllocateROI	Foundation
allocate_roi_vreg	InAllocateROIVReg	Full

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
allocate_static_array	InAllocateStaticArray	Technical
allocate_static_array_vreg	InAllocateStaticArrayVReg	Full
allocate_tuple	InAllocateTuple	Foundation
allocate_tuple_vreg	InAllocateTupleVReg	Full
allocate_value_bounds	InAllocateValueBounds	Scientific
allocate_value_bounds_vreg	InAllocateValueBoundsVReg	Full
allocate_vreg	InAllocateVReg	Full
allocate_vreg_vreg	InAllocateVRegVReg	Full
alpha_blend_constant	InAlphaBlendConstant	Foundation
alpha_blend_constant_vreg	InAlphaBlendConstantVReg	Full
alpha_blend_variable	InAlphaBlendVariable	Technical
alpha_blend_variable_vreg	InAlphaBlendVariableVReg	Full
array_to_lut	InArrayToLUT	Foundation
array_to_lut_vreg	InArrayToLUTVReg	Full
attach_iif_syntax_entity	InAttachEntity	Foundation
bind_match_point	InBindMatchPoint	Technical
bind_match_point_vreg	InBindMatchPointVReg	Full
bind_roi	InBindROI	Foundation
bind_roi_vreg	InBindROIVReg	Full
bit_shift	InBitShift	Foundation
bit_shift_vreg	InBitShiftVReg	Full
cartesian_to_polar	InCartesianToPolar	Scientific
cartesian_to_polar_vreg	InCartesianToPolarVReg	Full
chain_abort	InChainAbort	Full
chain_abort_vreg	InChainAbortVReg	Full
chain_begin	InChainBegin	Full
chain_delete	InChainDelete	Full
chain_delete_vreg	InChainDeleteVReg	Full
chain_end	InChainEnd	Full
chain_execute	InChainExecute	Full
chain_execute_vreg	InChainExecuteVReg	Full
chain_reload	InChainReload	Full
chain_reload_vreg	InChainReloadVReg	Full
classifier_bayes	InClassifierBayes	Scientific
classifier_bayes_vreg	InClassifierBayesVReg	Full
classifier_nearest_neighbour	InClassifierNearestNbr	Scientific
classifier_nearest_neighbour_vreg	InClassifierNearestNbrVReg	Full
close_iif_gateway	IvCloseGateway	Foundation
close_piks	IvClosePIKS	Foundation
close_piks_emergency	IvClosePIKSEmergency	Foundation
close_port	IvClosePort	Foundation
colour_conversion_linear	InColourConvLin	Foundation
colour_conversion_linear_vreg	InColourConvLinVReg	Full
colour_conversion_matrix	IvColourConvMatrix	Foundation
colour_conversion_matrix_vreg	IvColourConvMatrixVReg	Full
colour_conversion_nonlinear	InColourConvNonlin	Technical
colour_conversion_nonlinear_vreg	InColourConvNonlinVReg	Full

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
colour_conversion_subtractive	InColourConvSubtractive	Foundation
colour_conversion_subtractive_vreg	InColourConvSubtractiveVReg	Full
colour_lookup_interpolated	InColourLookupInterp	Technical
colour_lookup_interpolated_vreg	InColourLookupInterpVReg	Full
complement	InComplement	Foundation
complement_roi	InComplementROI	Technical
complement_roi_vreg	InComplementROIVReg	Full
complement_vreg	InComplementVReg	Full
complex_composition	InComplexComposition	Technical
complex_composition_vreg	InComplexCompositionVReg	Full
complex_conjugate	InComplexConjugate	Technical
complex_conjugate_vreg	InComplexConjugateVReg	Full
complex_decomposition	IvComplexDecomposition	Technical
complex_decomposition_vreg	IvComplexDecompositionVReg	Full
complex_magnitude	InComplexMagnitude	Technical
complex_magnitude_vreg	InComplexMagnitudeVReg	Full
composite_identifier_array_equal	IbIdArrayEqual	Scientific
composite_identifier_array_equal_vreg	InIdArrayEqualVReg	Full
composite_identifier_array_get	InIdArrayGet	Scientific
composite_identifier_array_get_vreg	InIdArrayGetVReg	Full
composite_identifier_array_put	InIdArrayPut	Scientific
composite_identifier_array_put_vreg	InIdArrayPutVReg	Full
composite_identifier_list_empty	IbIdListEmpty	Scientific
composite_identifier_list_empty_vreg	InIdListEmptyVReg	Full
composite_identifier_list_equal	IbIdListEqual	Scientific
composite_identifier_list_equal_vreg	InIdListEqualVReg	Full
composite_identifier_list_get	InIdListGet	Scientific
composite_identifier_list_get_vreg	InIdListGetVReg	Full
composite_identifier_list_insert	InIdListInsert	Scientific
composite_identifier_list_insert_vreg	InIdListInsertVReg	Full
composite_identifier_list_remove	InIdListRemove	Scientific
composite_identifier_list_remove_vreg	InIdListRemoveVReg	Full
composite_identifier_record_equal	IdIdRecordEqual	Scientific
composite_identifier_record_equal_vreg	InIdRecordEqualVReg	Full
composite_identifier_record_get	InIdRecordGet	Scientific
composite_identifier_record_get_vreg	InIdRecordGetVReg	Full
composite_identifier_record_put	InIdRecordPut	Scientific
composite_identifier_record_put_vreg	InIdRecordPutVReg	Full
compress_if_syntax_entity	InCompressEntity	Foundation
contsant_predicate	IbConstantPredicate	Technical
contsant_predicate_vreg	InConstantPredicateVReg	Full
convert_array_to_image	InConvertArrayToImage	Foundation
convert_array_to_image_vreg	InConvertArrayToImageVReg	Full
convert_image_datatype	InConvertImageDatatype	Foundation
convert_image_datatype_vreg	InConvertImageDatatypeVReg	Full
convert_image_to_array	InConvertImageToArray	Foundation
convert_image_to_array_vreg	InConvertImageToArrayVReg	Full

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
convert_image_to_roi	InConvertImageToROI	Technical
convert_image_to_roi_vreg	InConvertImageToROIVReg	Full
convert_roi_to_image	InConvertROIToImage	Foundation
convert_roi_to_image_vreg	InConvertROIToImageVReg	Full
convolve_2d	InConvolve2D	Foundation
convolve_2d_vreg	InConvolve2DVReg	Full
convolve_5d	InConvolve5D	Technical
convolve_5d_vreg	InConvolve5DVReg	Full
copy_iif_syntax_entity	InCopyEntity	Foundation
copy_window	InCopyWindow	Foundation
copy_window_roi	InCopyWindowROI	Technical
copy_window_roi_vreg	InCopyWindowROIVReg	Full
copy_window_vreg	InCopyWindowVReg	Full
create_iif_syntax_entity	InCreateEntity	Foundation
create_tuple	InCreateTuple	Foundation
create_tuple_vreg	InCreateTupleVReg	Full
cross_correlation	InCrossCorrelation	Scientific
cross_correlation_vreg	InCrossCorrelationVReg	Full
deallocate_data_object	IvDeallocateDataObject	Foundation
deallocate_data_object_vreg	IvDeallocateDataObjectVReg	Full
decompress_iif_syntax_entity	IstDecompressEntity	Foundation
define_sub_image	InDefineSubImage	Foundation
define_sub_image_vreg	InDefineSubImageVReg	Full
delete_iif_atomic_sequence_element	InDeleteAtomicSeqElement	Foundation
delete_iif_syntax_entity	IvDeleteEntity	Foundation
detach_iif_syntax_entity	InDetachEntity	Foundation
difference_measures	IrDiffMeasures	Technical
difference_measures_vreg	InDiffMeasuresVReg	Full
diffuse	InDiffuse	Foundation
diffuse_vreg	InDiffuseVReg	Full
dither	InDither	Foundation
dither_vreg	InDitherVReg	Full
draw_pixels	InDrawPixels	Foundation
draw_pixels_roi	InDrawPixelsROI	Technical
draw_pixels_roi_vreg	InDrawPixelsROIVReg	Full
draw_pixels_vreg	InDrawPixelsVReg	Full
dyadic_arithmetic	InDyadicArith	Foundation
dyadic_arithmetic_vreg	InDyadicArithVReg	Full
dyadic_complex	InDyadicComplex	Technical
dyadic_complex_vreg	InDyadicComplexVReg	Full
dyadic_logical	InDyadicLogical	Foundation
dyadic_logical_roi	InDyadicLogicalROI	Technical
dyadic_logical_roi_vreg	InDyadicLogicalROIVReg	Full
dyadic_logical_vreg	InDyadicLogicalVReg	Full
dyadic_predicate	InDyadicPredicate	Foundation
dyadic_predicate_vreg	InDyadicPredicateVReg	Full
edge_detection_orthogonal_gradient	InEdgeDetectOrthGrad	Scientific

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
edge_detection_orthogonal_gradient_vreg	InEdgeDetectOrthGradVReg	Full
edge_detection_second_derivative	InEdgeDetect2ndDeriv	Scientific
edge_detection_second_derivative_vreg	InEdgeDetect2ndDerivVReg	Full
edge_detection_template_gradient	InEdgeDetectTemplateGrad	Scientific
edge_detection_template_gradient_vreg	InEdgeDetectTemplateGradVReg	Full
equal_predicate	IbEqualPredicate	Technical
equal_predicate_vreg	InEqualPredicateVReg	Full
erosion_dilation_boolean	InErosionDilateBool	Technical
erosion_dilation_boolean_roi	InErosionDilateBoolROI	Technical
erosion_dilation_boolean_roi_vreg	InErosionDilateBoolROIVReg	Full
erosion_dilation_boolean_vreg	InErosionDilateBoolVReg	Full
erosion_dilation_grey	InErosionDilateGrey	Technical
erosion_dilation_grey_vreg	InErosionDilateGreyVReg	Full
error_function_scaling	InErrorFunctionScaling	Scientific
error_function_scaling_vreg	InErrorFunctionScalingVReg	Full
error_handler	IvErrorHandler	Foundation
error_logger	IvErrorLogger	Foundation
error_test	IbErrorTest	Foundation
error_test_vreg	InErrorTestVReg	Full
export_histogram	ItuaExportHist	Foundation
export_iif_data_to_piks	InExportIIFDataToPIKS	Foundation - with PIKS only
export_image	IstExportImage	Foundation
export_lut	IstExportLUT	Foundation
export_matrix	IstExportMatrix	Foundation
export_neighbourhood_array	IstExportNbhoodArray	Foundation
export_roi_array	ItbaExportROIArray	Technical
export_static_array	IstExportStaticArray	Technical
export_tuple	IstExportTuple	Foundation
export_value_bounds	IstExportValueBounds	Scientific
extract_pixel_plane	InExtractPixelPlane	Foundation
extract_pixel_plane_roi	InExtractPixelPlaneROI	Technical
extract_pixel_plane_roi_vreg	InExtractPixelPlaneROIVReg	Full
extract_pixel_plane_vreg	InExtractPixelPlaneVReg	Full
extrema	IvExtrema	Foundation
false_colour	InFalseColour	Scientific
false_colour_vreg	InFalseColourVReg	Full
fill_region	InFillRegion	Technical
fill_region_roi	InFillRegionROI	Technical
fill_region_roi_vreg	InFillRegionROIVReg	Full
fill_region_vreg	InFillRegionVReg	Full
filter_butterworth	InFilterButterworth	Technical
filter_butterworth_vreg	InFilterButterworthVReg	Full
filter_gaussian	InFilterGauss	Technical
filter_gaussian_vreg	InFilterGaussVReg	Full
filter_inverse	InFilterInverse	Scientific
filter_inverse_vreg	InFilterInverseVReg	Full
filter_matched	InFilterMatched	Scientific

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
filter_matched_vreg	InFilterMatchedVReg	Full
filter_wiener	InFilterWiener	Scientific
filter_wiener_vreg	InFilterWienerVReg	Full
filter_zonal	InFilterZonal	Technical
filter_zonal_vreg	InFilterZonalVReg	Full
filtering_homomorphic	InFilteringHomomorphic	Scientific
filtering_homomorphic_vreg	InFilteringHomomorphicVReg	Full
filtering_linear	InFilteringLin	Technical
filtering_linear_vreg	InFilteringLinVReg	Full
filtering_median	InFilteringMedian	Technical
filtering_median_vreg	InFilteringMedianVReg	Full
filtering_pseudomedian	InFilteringPseudomedian	Scientific
filtering_pseudomedian_vreg	InFilteringPseudomedianVReg	Full
filtering_rank_order	InFilteringRankOrder	Scientific
filtering_rank_order_vreg	InFilteringRankOrderVReg	Full
flip_spin_transpose	InFlipSpinTranspose	Foundation
flip_spin_transpose_roi	InFlipSpinTransposeROI	Foundation
flip_spin_transpose_roi_vreg	InFlipSpinTransposeROIIVReg	Full
flip_spin_transpose_vreg	InFlipSpinTransposeVReg	Full
gamma_correction	InGammaCorrection	Technical
gamma_correction_vreg	InGammaCorrectionVReg	Full
generate_and_write_iif_image	InGenerateWriteImage	Foundation
get_colour_pixel	IvGetColourPixel	Foundation
get_colour_pixel_vreg	IvGetColourPixelVReg	Full
get_iif_entity_component	InGetEntityCmp	Foundation
get_iif_syntax_entity_value	IstGetEntityValue	Foundation
get_image_subrange	InGetImageSubrange	Foundation
get_pixel	IstGetPixel	Foundation
get_pixel_array	IstGetPixelArray	Foundation
get_pixel_array_roi	ItbaGetPixelArrayROI	Foundation
get_pixel_array_roi_vreg	InGetPixelArrayROIIVReg	Full
get_pixel_record	IstGetPixelRecord	Technical
get_pixel_record_roi	ItbaGetPixelRecordROI	Technical
get_pixel_record_roi_vreg	InGetPixelRecordROIIVReg	Full
get_pixel_record_vreg	InGetPixelVReg	Full
get_pixel_roi	IbGetPixelROI	Foundation
histogram_1d	InHist1D	Foundation
histogram_1d_vreg	InHist1DVReg	Full
histogram_2d	InHist2D	Scientific
histogram_2d_vreg	InHist2DVReg	Full
histogram_modification	InHistModification	Scientific
histogram_modification_vreg	InHistModificationVReg	Full
histogram_scaling	InHistScaling	Scientific
histogram_scaling_vreg	InHistScalingVReg	Full
histogram_shape_1d	ItraHistShape1D	Scientific
histogram_shape_2d	ItraHistShape2D	Scientific
hit_or_miss	InHitMiss	Technical
hit_or_miss_roi	InHitMissROI	Technical

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
hit_or_miss_roi_vreg	InHitMissROIVReg	Full
hit_or_miss_vreg	InHitMissVReg	Full
hough_transform	InHoughTransform	Scientific
hough_transform_vreg	InHoughTransformVReg	Full
image_bar_chart	InImageBarChart	Technical
image_bar_chart_vreg	InImageBarChartVReg	Full
image_constant	InImageConstant	Foundation
image_constant_vreg	InImageConstantVReg	Full
image_gaussian	InImageGauss	Scientific
image_gaussian_vreg	InImageGaussVReg	Full
image_grey_scale	InImageGreyScale	Technical
image_grey_scale_vreg	InImageGreyScaleVReg	Full
image_random_number	InImageRandomNumber	Scientific
image_random_number_vreg	InImageRandomNumberVReg	Full
import_histogram	InImportHist	Foundation
import_histogram_vreg	InImportHistVReg	Full
import_iif_data_from_piks	InImportIIFDataFromPIKS	Foundation - with PIKS only
import_image	InImportImage	Foundation
import_image_vreg	InImportImageVReg	Full
import_lut	InImportLUT	Foundation
import_lut_vreg	InImportLUTVReg	Full
import_matrix	InImportMatrix	Foundation
import_matrix_vreg	InImportMatrixVReg	Full
import_neighbourhood_array	InImportNbhoodArray	Foundation
import_neighbourhood_array_vreg	InImportNbhoodArrayVReg	Full
import_roi_array	InImportROIArray	Technical
import_roi_array_vreg	InImportROIArrayVReg	Full
import_static_array	InImportStaticArray	Technical
import_static_array_vreg	InImportStaticArrayVReg	Full
import_tuple	InImportTuple	Foundation
import_tuple_vreg	InImportTupleVReg	Full
import_value_bounds	InImportValueBounds	Scientific
import_value_bounds_vreg	InImportValueBoundsVReg	Full
impulse_boxcar	InImpulseBoxcar	Scientific
impulse_boxcar_vreg	InImpulseBoxcarVReg	Full
impulse_derivative_of_gaussian	InImpulseDerivGauss	Scientific
impulse_derivative_of_gaussian_vreg	InImpulseDerivGaussVReg	Full
impulse_difference_of_gaussians	InImpulseDiffGauss	Scientific
impulse_difference_of_gaussians_vreg	InImpulseDiffGaussVReg	Full
impulse_elliptical	InImpulseElliptical	Technical
impulse_elliptical_vreg	InImpulseEllipticalVReg	Full
impulse_gaussian	InImpulseGauss	Technical
impulse_gaussian_vreg	InImpulseGaussVReg	Full
impulse_laplacian_of_gaussian	InImpulseLaplaceGauss	Scientific
impulse_laplacian_of_gaussian_vreg	InImpulseLaplaceGaussVReg	Full
impulse_pyramid	InImpulsePyramid	Scientific
impulse_pyramid_vreg	InImpulsePyramidVReg	Full

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
impulse_rectangular	InImpulseRectangular	Foundation
impulse_rectangular_vreg	InImpulseRectangularVReg	Full
impulse_sinc	InImpulseSinc	Scientific
impulse_sinc_vreg	InImpulseSincVReg	Full
input_object	IsnInputObject	Foundation - with IIF only
input_object_vreg	InInputObjectVReg	Full - with IIF only
inquire_chain_environment	IvInquireChainEnvironment	Full
inquire_chain_status	IvInquireChainStatus	Full
inquire_elements	IvInquireElements	Foundation
inquire_iif_gateway_status	IvInquireGatewayStatus	Foundation
inquire_iif_syntax_component_type_definition	IvCmpTypeDef	Foundation
inquire_iif_syntax_entity_choice	IvInquireEntityChc	Foundation
inquire_iif_syntax_entity_number	IvInquireEntityNumber	Foundation
inquire_iif_syntax_entity_optional	IvInquireEntityOptional	Foundation
inquire_iif_syntax_entity_sequence	IbInquireEntitySeq	Foundation
inquire_iif_syntax_entity_type_definition	IvInquireEntityTypeDef	Foundation
inquire_iif_syntax_entity_validity	IvInquireEntityValidity	Foundation
inquire_image	IvInquireImage	Foundation
inquire_index_assignment	IvInquireIndexAssignment	Technical
inquire_non_image_object	IvInquireNonImageObject	Foundation
inquire_piks_implementation	IvInquirePIKSImp	Foundation
inquire_piks_status	IvInquirePIKSStatus	Foundation
inquire_port	IvInquirePort	Foundation
inquire_repository	IvInquireRepository	Foundation
inquire_resampling	IvInquireResampling	Foundation
inquire_supported_functionality	IvInquireFunctionality	Foundation
insert_pixel_plane	InInsertPixelPlane	Foundation
insert_pixel_plane_roi	InInsertPixelPlaneROI	Technical
insert_pixel_plane_roi_vreg	InInsertPixelPlaneROIVReg	Full
insert_pixel_plane_vreg	InInsertPixelPlaneVReg	Full
label_objects	InLabelObjects	Scientific
label_objects_vreg	InLabelObjectsVReg	Full
laws_texture_features	InLawsTextureFeatures	Scientific
laws_texture_features_vreg	InLawsTextureFeaturesVReg	Full
level_slice	InLevelSlice	Scientific
level_slice_vreg	InLevelSliceVReg	Full
line_profile	InLineProfile	Scientific
line_profile_vreg	ItraLineProfileVReg	Full
lookup	InLookup	Foundation
lookup_interpolated	InLookupInterp	Technical
lookup_interpolated_vreg	InLookupInterpVReg	Full
lookup_vreg	InLookupVReg	Full
luminance_generation	InLumGeneration	Technical
luminance_generation_vreg	InLumGenerationVReg	Full
markov_spectrum	InMarkovSpectrum	Scientific
markov_spectrum_vreg	InMarkovSpectrumVReg	Full
moments	IvMoments	Foundation

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
moments_vreg	IvMomentsVReg	Full
monadic_arithmetic	InMonadicArith	Foundation
monadic_arithmetic_vreg	InMonadicArithVReg	Full
monadic_complex	InMonadicComplex	Technical
monadic_complex_vreg	InMonadicComplexVReg	Full
monadic_logical	InMonadicLogical	Foundation
monadic_logical_roi	InMonadicLogicalROI	Technical
monadic_logical_roi_vreg	InMonadicLogicalROIIVReg	Full
monadic_logical_vreg	InMonadicLogicalVReg	Full
morphic_processor	InMorphicProcessor	Foundation
morphic_processor_roi	InMorphicProcessorROI	Technical
morphic_processor_roi_vreg	InMorphicProcessorROIIVReg	Full
morphic_processor_vreg	InMorphicProcessorVReg	Full
morphology	InMorphology	Technical
morphology_roi	InMorphologyROI	Technical
morphology_roi_vreg	InMorphologyROIIVReg	Full
morphology_vreg	InMorphologyVReg	Full
neighbour_count	InNbrCount	Scientific
neighbour_count_vreg	InNbrCountVReg	Full
noise_combination	InNoiseCombination	Scientific
noise_combination_vreg	InNoiseCombinationVReg	Full
open_close	InOpenClose	Scientific
open_close_roi	InOpenCloseROI	Scientific
open_close_roi_vreg	InOpenCloseROIIVReg	Full
open_close_vreg	InOpenCloseVReg	Full
open_iif_gateway	IvOpenGateway	Foundation
open_piks	IvOpenPIKS	Foundation
open_port	InOpenPort	Foundation
outlier_removal	InOutlierRemoval	Technical
outlier_removal_vreg	InOutlierRemovalVReg	Full
output_object	InOutputObject	Foundation - with IIF only
paint_pixels	InPaintPixels	Technical
paint_pixels_vreg	InPaintPixelsVReg	Full
perimeter_code_generator	ItuaPerimeterCodeGenerator	Scientific
polar_to_cartesian	InPolarToCartesian	Scientific
polar_to_cartesian_vreg	InPolarToCartesianVReg	Full
power law_scaling	InPowerLawScaling	Technical
power law_scaling_vreg	InPowerLawScalingVReg	Full
pseudocolour	InPseudocolour	Technical
pseudocolour_vreg	InPseudocolourVReg	Full
put_colour_pixel	InPutColourPixel	Foundation
put_colour_pixel_vreg	InPutColourPixelVReg	Full
put_iif_syntax_entity_value	InPutEntityValue	Foundation
put_image_subrange	ItzPutImageSubrange	Foundation
put_pixel	InPutPixel	Foundation
put_pixel_array	InPutPixelArray	Foundation
put_pixel_array_roi	InPutPixelArrayROI	Technical

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
put_pixel_array_roi_vreg	InPutPixelArrayROIvReg	Full
put_pixel_array_vreg	InPutPixelArrayVReg	Full
put_pixel_record	InPutPixelRecord	Technical
put_pixel_record_roi	InPutPixelRecordROI	Technical
put_pixel_record_roi_vreg	InPutPixelRecordROIvReg	Full
put_pixel_record_vreg	InPutPixelRecordVReg	Full
put_pixel_roi	InPutPixelROI	Technical
put_pixel_roi_vreg	InPutPixelROIvReg	Full
put_pixel_vreg	InPutPixelVReg	Full
read_and_parse_iif_image	InReadParseImage	Foundation
release_iif_syntax_entity_identifier	IvReleaseEntityId	Foundation
rescale	InRescale	Foundation
rescale_roi	InRescaleROI	Foundation
rescale_roi_vreg	InRescaleROIvReg	Full
rescale_vreg	InRescaleVReg	Full
reset_iif_gateway	IvResetGateway	Foundation
reset_port	IvResetPort	Foundation
resize	InResize	Foundation
resize_roi	InResizeROI	Foundation
resize_roi_vreg	InResizeROIvReg	Full
resize_vreg	InResizeVReg	Full
return_repository_id	InReturnRepositoryId	Foundation
return_repository_id_vreg	InReturnRepositoryIdVReg	Full
roi_coordinate_array	InROICoordinateArray	Technical
roi_coordinate_array_vreg	InROICoordinateArrayVReg	Full
roi_elliptical	InROIElliptical	Technical
roi_elliptical_vreg	InROIEllipticalVReg	Full
roi_polygon	InROIPolygon	Technical
roi_polygon_vreg	InROIPolygonVReg	Full
roi_rectangular	InROIRectangular	Foundation
roi_rectangular_vreg	InROIRectangularVReg	Full
rotate	InRotate	Foundation
rotate_roi	InRotateROI	Technical
rotate_roi_vreg	InRotateROIvReg	Full
rotate_vreg	InRotateVReg	Full
rubber_band_scaling	InRubberBandScaling	Scientific
rubber_band_scaling_vreg	InRubberBandScalingVReg	Full
sequence_average	InSeqAverage	Scientific
sequence_average_vreg	InSeqAverageVReg	Full
sequence_karhunen_loeve	InSeqKarhunenLoeve	Scientific
sequence_karhunen_loeve_vreg	InSeqKarhunenLoeveVReg	Full
sequence_running_measures	InSeqRunningMeasures	Scientific
sequence_running_measures_vreg	InSeqRunningMeasuresVReg	Full
set_error_handler	ItfSetErrorHandler	Foundation
set_globals	IvSetGlobals	Foundation
set_globals_vreg	IvSetGlobalsVReg	Full
set_image_attributes	InSetImageAttrs	Foundation

Table 3 - Function names alphabetical by function name (continued)

Function Name	Bound Name	Least Profile
set_image_attributes_vreg	InSetImageAttrsVReg	Full
set_index_assignment	IvSetIndexAssignment	Technical
set_index_assignment_vreg	IvSetIndexAssignmentVReg	Full
shape_metrics	ItraShapeMetrics	Scientific
spatial_moments_invariant	ItraSpatialMomentsInvariant	Scientific
spatial_moments_scaled	ItraSpatialMomentsScaled	Scientific
split_image	InSplitImage	Foundation
split_image_vreg	InSplitImageVReg	Full
subsample	InSubsample	Foundation
subsample_roi	InSubsampleROI	Technical
subsample_roi_vreg	InSubsampleROIIVReg	Full
subsample_vreg	InSubsampleVReg	Full
synchronize	IvSynch	Scientific
template_match	InTemplMatch	Scientific
template_match_vreg	InTemplMatchVReg	Full
threshold	InThreshold	Foundation
threshold_vreg	InThresholdVReg	Full
transform_cosine	InTransformCosine	Technical
transform_cosine_vreg	InTransformCosineVReg	Full
transform_fourier	InTransformFourier	Technical
transform_fourier_vreg	InTransformFourierVReg	Full
transform_hadamard	InTransformHadamard	Scientific
transform_hadamard_vreg	InTransformHadamardVReg	Full
transform_hartley	InTransformHartley	Scientific
transform_hartley_vreg	InTransformHartleyVReg	Full
translate	InTranslate	Foundation
translate_roi	InTranslateROI	Foundation
translate_roi_vreg	InTranslateROIIVReg	Full
translate_vreg	InTranslateVReg	Full
unary_integer	InUnaryInteger	Foundation
unary_integer_vreg	InUnaryIntegerVReg	Full
unary_real	InUnaryReal	Technical
unary_real_vreg	InUnaryRealVReg	Full
unsharp_mask	InUnsharpMask	Technical
unsharp_mask_vreg	InUnsharpMaskVReg	Full
value_bounds	InValueBounds	Scientific
value_bounds_vreg	InValueBoundsVReg	Full
vreg_alter	InVRegAlter	Full
vreg_alter_vreg	InVRegAlterVReg	Full
vreg_clear	InVRegClear	Full
vreg_clear_vreg	InVRegClearVReg	Full
vreg_conditional	IvVRegConditional	Full
vreg_conditional_vreg	IvVRegConditionalVReg	Full
vreg_copy	InVRegCopy	Full
vreg_copy_vreg	InVRegCopyVReg	Full
vreg_delete	InVRegDelete	Full
vreg_delete_vreg	InVRegDeleteVReg	Full

Table 3 - Function names alphabetical by function name (concluded)

Function Name	Bound Name	Least Profile
vreg_get	lvVRegGet	Full
vreg_set	lnVRegSet	Full
vreg_set_vreg	lnVRegSetVReg	Full
vreg_wait	lnVRegWait	Full
vreg_wait_vreg	lnVRegWaitVReg	Full
wallis_statistical_differencing	lnWallisStatDiff	Technical
wallis_statistical_differencing_vreg	lnWallisStatDiffVReg	Full
warp_control_point	lnWarpControlPoint	Technical
warp_control_point_roi	lnWarpControlPointROI	Technical
warp_control_point_roi_vreg	lnWarpControlPointROIIVReg	Full
warp_control_point_vreg	lnWarpControlPointVReg	Full
warp_lut	lnWarpLUT	Technical
warp_lut_roi	lnWarpLUTROI	Technical
warp_lut_roi_vreg	lnWarpLUTROIIVReg	Full
warp_lut_vreg	lnWarpLUTVReg	Full
warp_polynomial	lnWarpPolynomial	Technical
warp_polynomial_roi	lnWarpPolynomialROI	Technical
warp_polynomial_roi_vreg	lnWarpPolynomialROIIVReg	Full
warp_polynomial_vreg	lnWarpPolynomialVReg	Full
window_level	lnWindowLevel	Foundation
window_level_vreg	lnWindowLevelVReg	Full
window_statistics	lnWindowStatistics	Scientific
window_statistics_vreg	lnWindowStatisticsVReg	Full
windowing_function	lnWindowingFunction	Scientific
windowing_function_vreg	lnWindowingFunctionVReg	Full
z_merge	lnZMerge	Technical
z_merge_vreg	lnZMergeVReg	Full
zoom	lnZoom	Foundation
zoom_roi	lnZoomROI	Foundation
zoom_roi_vreg	lnZoomROIIVReg	Full
zoom_vreg	lnZoomVReg	Full
3d_slice	ln3DSlice	Scientific
3d_slice_vreg	ln3DSliceVReg	Full

4.2.2 Alphabetical by bound name

Table 4 - Function names alphabetical by bound name

Bound Name	Function Name	Least Profile
IbConstantPredicate	contsant_predicate	Technical
IbEqualPredicate	equal_predicate	Technical
IbErrorTest	error_test	Foundation
IbGetPixelROI	get_pixel_roi	Foundation
IbIdArrayEqual	composite_identifier_array_equal	Scientific
IbIdListEmpty	composite_identifier_list_empty	Scientific
IbIdListEqual	composite_identifier_list_equal	Scientific
IbInquireEntitySeq	inquire_iif_syntax_entity_sequence	Foundation
IdIdRecordEqual	composite_identifier_record_equal	Scientific
InAdaptiveHistEqual	adaptive_histogram_equalization	Scientific
InAdaptiveHistEqualVReg	adaptive_histogram_equalization_vreg	Full
InAllocateChain	allocate_chain	Full
InAllocateChainVReg	allocate_chain_vreg	Full
InAllocateHist	allocate_histogram	Foundation
InAllocateHistVReg	allocate_histogram_vreg	Full
InAllocateIdArray	allocate_composite_identifier_array	Scientific
InAllocateIdArrayVReg	allocate_composite_identifier_array_vreg	Full
InAllocateIdList	allocate_composite_identifier_list	Scientific
InAllocateIdListVReg	allocate_composite_identifier_list_vreg	Full
InAllocateIdRecord	allocate_composite_identifier_record	Scientific
InAllocateIdRecordVReg	allocate_composite_identifier_record_vreg	Full
InAllocateImage	allocate_image	Foundation
InAllocateImageVReg	allocate_image_vreg	Full
InAllocateLUT	allocate_lookup_table	Foundation
InAllocateLUTVReg	allocate_lookup_table_vreg	Full
InAllocateMatrix	allocate_matrix	Foundation
InAllocateMatrixVReg	allocate_matrix_vreg	Full
InAllocateNbhoodArray	allocate_neighbourhood_array	Foundation
InAllocateNbhoodArrayVReg	allocate_neighbourhood_array_vreg	Full
InAllocatePixelRecord	allocate_pixel_record	Technical
InAllocatePixelRecordVReg	allocate_pixel_record_vreg	Full
InAllocateROI	allocate_roi	Foundation
InAllocateROIVReg	allocate_roi_vreg	Full
InAllocateStaticArray	allocate_static_array	Technical
InAllocateStaticArrayVReg	allocate_static_array_vreg	Full
InAllocateTuple	allocate_tuple	Foundation
InAllocateTupleVReg	allocate_tuple_vreg	Full
InAllocateValueBounds	allocate_value_bounds	Scientific
InAllocateValueBoundsVReg	allocate_value_bounds_vreg	Full
InAllocateVReg	allocate_vreg	Full
InAllocateVRegVReg	allocate_vreg_vreg	Full
InAlphaBlendConstant	alpha_blend_constant	Foundation
InAlphaBlendConstantVReg	alpha_blend_constant_vreg	Full
InAlphaBlendVariable	alpha_blend_variable	Technical
InAlphaBlendVariableVReg	alpha_blend_variable_vreg	Full

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InArrayToLUT	array_to_lut	Foundation
InArrayToLUTVReg	array_to_lut_vreg	Full
InAttachEntity	attach_iif_syntax_entity	Foundation
InBindMatchPoint	bind_match_point	Technical
InBindMatchPointVReg	bind_match_point_vreg	Full
InBindROI	bind_roi	Foundation
InBindROIVReg	bind_roi_vreg	Full
InBitShift	bit_shift	Foundation
InBitShiftVReg	bit_shift_vreg	Full
InCartesianToPolar	cartesian_to_polar	Scientific
InCartesianToPolarVReg	cartesian_to_polar_vreg	Full
InChainAbort	chain_abort	Full
InChainAbortVReg	chain_abort_vreg	Full
InChainBegin	chain_begin	Full
InChainDelete	chain_delete	Full
InChainDeleteVReg	chain_delete_vreg	Full
InChainEnd	chain_end	Full
InChainExecute	chain_execute	Full
InChainExecuteVReg	chain_execute_vreg	Full
InChainReload	chain_reload	Full
InChainReloadVReg	chain_reload_vreg	Full
InClassifierBayes	classifier_bayes	Scientific
InClassifierBayesVReg	classifier_bayes_vreg	Full
InClassifierNearestNbr	classifier_nearest_neighbour	Scientific
InClassifierNearestNbrVReg	classifier_nearest_neighbour_vreg	Full
InColourConvLin	colour_conversion_linear	Foundation
InColourConvLinVReg	colour_conversion_linear_vreg	Full
InColourConvNonlin	colour_conversion_nonlinear	Technical
InColourConvNonlinVReg	colour_conversion_nonlinear_vreg	Full
InColourConvSubtractive	colour_conversion_subtractive	Foundation
InColourConvSubtractiveVReg	colour_conversion_subtractive_vreg	Full
InColourLookupInterp	colour_lookup_interpolated	Technical
InColourLookupInterpVReg	colour_lookup_interpolated_vreg	Full
InComplement	complement	Foundation
InComplementROI	complement_roi	Technical
InComplementROIVReg	complement_roi_vreg	Full
InComplementVReg	complement_vreg	Full
InComplexComposition	complex_composition	Technical
InComplexCompositionVReg	complex_composition_vreg	Full
InComplexConjugate	complex_conjugate	Technical
InComplexConjugateVReg	complex_conjugate_vreg	Full
InComplexMagnitude	complex_magnitude	Technical
InComplexMagnitudeVReg	complex_magnitude_vreg	Full
InCompressEntity	compress_iif_syntax_entity	Foundation
InConstantPredicateVReg	constant_predicate_vreg	Full
InConvertArrayToImage	convert_array_to_image	Foundation
InConvertArrayToImageVReg	convert_array_to_image_vreg	Full

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InConvertImageDatatype	convert_image_datatype	Foundation
InConvertImageDatatypeVReg	convert_image_datatype_vreg	Full
InConvertImageToArray	convert_image_to_array	Foundation
InConvertImageToArrayVReg	convert_image_to_array_vreg	Full
InConvertImageToROI	convert_image_to_roi	Technical
InConvertImageToROIVReg	convert_image_to_roi_vreg	Full
InConvertROIToImage	convert_roi_to_image	Foundation
InConvertROIToImageVReg	convert_roi_to_image_vreg	Full
InConvolve2D	convolve_2d	Foundation
InConvolve2DVReg	convolve_2d_vreg	Full
InConvolve5D	convolve_5d	Technical
InConvolve5DVReg	convolve_5d_vreg	Full
InCopyEntity	copy_iif_syntax_entity	Foundation
InCopyWindow	copy_window	Foundation
InCopyWindowROI	copy_window_roi	Technical
InCopyWindowROIVReg	copy_window_roi_vreg	Full
InCopyWindowVReg	copy_window_vreg	Full
InCreateEntity	create_iif_syntax_entity	Foundation
InCreateTuple	create_tuple	Foundation
InCreateTupleVReg	create_tuple_vreg	Full
InCrossCorrelation	cross_correlation	Scientific
InCrossCorrelationVReg	cross_correlation_vreg	Full
InDefineSubImage	define_sub_image	Foundation
InDefineSubImageVReg	define_sub_image_vreg	Full
InDeleteAtomicSeqElement	delete_iif_atomic_sequence_element	Foundation
InDetachEntity	detach_iif_syntax_entity	Foundation
InDiffMeasuresVReg	difference_measures_vreg	Full
InDiffuse	diffuse	Foundation
InDiffuseVReg	diffuse_vreg	Full
InDither	dither	Foundation
InDitherVReg	dither_vreg	Full
InDrawPixels	draw_pixels	Foundation
InDrawPixelsROI	draw_pixels_roi	Technical
InDrawPixelsROIVReg	draw_pixels_roi_vreg	Full
InDrawPixelsVReg	draw_pixels_vreg	Full
InDyadicArith	dyadic_arithmetic	Foundation
InDyadicArithVReg	dyadic_arithmetic_vreg	Full
InDyadicComplex	dyadic_complex	Technical
InDyadicComplexVReg	dyadic_complex_vreg	Full
InDyadicLogical	dyadic_logical	Foundation
InDyadicLogicalROI	dyadic_logical_roi	Technical
InDyadicLogicalROIVReg	dyadic_logical_roi_vreg	Full
InDyadicLogicalVReg	dyadic_logical_vreg	Full
InDyadicPredicate	dyadic_predicate	Foundation
InDyadicPredicateVReg	dyadic_predicate_vreg	Full
InEdgeDetectOrthGrad	edge_detection_orthogonal_gradient	Scientific
InEdgeDetectOrthGradVReg	edge_detection_orthogonal_gradient_vreg	Full

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InEdgeDetect2ndDeriv	edge_detection_second_derivative	Scientific
InEdgeDetect2ndDerivVReg	edge_detection_second_derivative_vreg	Full
InEdgeDetectTemplateGrad	edge_detection_template_gradient	Scientific
InEdgeDetectTemplateGradVReg	edge_detection_template_gradient_vreg	Full
InEqualPredicateVReg	equal_predicate_vreg	Full
InErosionDilateBool	erosion_dilation_boolean	Technical
InErosionDilateBoolROI	erosion_dilation_boolean_roi	Technical
InErosionDilateBoolROIVReg	erosion_dilation_boolean_roi_vreg	Full
InErosionDilateBoolVReg	erosion_dilation_boolean_vreg	Full
InErosionDilateGrey	erosion_dilation_grey	Technical
InErosionDilateGreyVReg	erosion_dilation_grey_vreg	Full
InErrorFunctionScaling	error_function_scaling	Scientific
InErrorFunctionScalingVReg	error_function_scaling_vreg	Full
InErrorTestVReg	error_test_vreg	Full
InExportIIFDataToPIKS	export_iif_data_to_piks	Foundation - with PIKS only
InExtractPixelPlane	extract_pixel_plane	Foundation
InExtractPixelPlaneROI	extract_pixel_plane_roi	Technical
InExtractPixelPlaneROIVReg	extract_pixel_plane_roi_vreg	Full
InExtractPixelPlaneVReg	extract_pixel_plane_vreg	Full
InFalseColour	false_colour	Scientific
InFalseColourVReg	false_colour_vreg	Full
InFillRegion	fill_region	Technical
InFillRegionROI	fill_region_roi	Technical
InFillRegionROIVReg	fill_region_roi_vreg	Full
InFillRegionVReg	fill_region_vreg	Full
InFilterButterworth	filter_butterworth	Technical
InFilterButterworthVReg	filter_butterworth_vreg	Full
InFilterGauss	filter_gaussian	Technical
InFilterGaussVReg	filter_gaussian_vreg	Full
InFilteringHomomorphic	filtering_homomorphic	Scientific
InFilteringHomomorphicVReg	filtering_homomorphic_vreg	Full
InFilteringLin	filtering_linear	Technical
InFilteringLinVReg	filtering_linear_vreg	Full
InFilteringMedian	filtering_median	Technical
InFilteringMedianVReg	filtering_median_vreg	Full
InFilteringPseudomedian	filtering_pseudomedian	Scientific
InFilteringPseudomedianVReg	filtering_pseudomedian_vreg	Full
InFilteringRankOrder	filtering_rank_order	Scientific
InFilteringRankOrderVReg	filtering_rank_order_vreg	Full
InFilterInverse	filter_inverse	Scientific
InFilterInverseVReg	filter_inverse_vreg	Full
InFilterMatched	filter_matched	Scientific
InFilterMatchedVReg	filter_matched_vreg	Full
InFilterWiener	filter_wiener	Scientific
InFilterWienerVReg	filter_wiener_vreg	Full
InFilterZonal	filter_zonal	Technical
InFilterZonalVReg	filter_zonal_vreg	Full

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InFlipSpinTranspose	flip_spin_transpose	Foundation
InFlipSpinTransposeROI	flip_spin_transpose_roi	Foundation
InFlipSpinTransposeROIvReg	flip_spin_transpose_roi_vreg	Full
InFlipSpinTransposevReg	flip_spin_transpose_vreg	Full
InGammaCorrection	gamma_correction	Technical
InGammaCorrectionvReg	gamma_correction_vreg	Full
InGenerateWriteImage	generate_and_write_iif_image	Foundation
InGetEntityCmp	get_iif_entity_component	Foundation
InGetImageSubrange	get_image_subrange	Foundation
InGetPixelArrayROIvReg	get_pixel_array_roi_vreg	Full
InGetPixelRecordROIvReg	get_pixel_record_roi_vreg	Full
InGetPixelvReg	get_pixel_record_vreg	Full
InHist1D	histogram_1d	Foundation
InHist1DVReg	histogram_1d_vreg	Full
InHist2D	histogram_2d	Scientific
InHist2DVReg	histogram_2d_vreg	Full
InHistModification	histogram_modification	Scientific
InHistModificationvReg	histogram_modification_vreg	Full
InHistScaling	histogram_scaling	Scientific
InHistScalingvReg	histogram_scaling_vreg	Full
InHitMiss	hit_or_miss	Technical
InHitMissROI	hit_or_miss_roi	Technical
InHitMissROIvReg	hit_or_miss_roi_vreg	Full
InHitMissvReg	hit_or_miss_vreg	Full
InHoughTransform	hough_transform	Scientific
InHoughTransformvReg	hough_transform_vreg	Full
InIdArrayEqualvReg	composite_identifier_array_equal_vreg	Full
InIdArrayGet	composite_identifier_array_get	Scientific
InIdArrayGetvReg	composite_identifier_array_get_vreg	Full
InIdArrayPut	composite_identifier_array_put	Scientific
InIdArrayPutvReg	composite_identifier_array_put_vreg	Full
InIdListEmptyvReg	composite_identifier_list_empty_vreg	Full
InIdListEqualvReg	composite_identifier_list_equal_vreg	Full
InIdListGet	composite_identifier_list_get	Scientific
InIdListGetvReg	composite_identifier_list_get_vreg	Full
InIdListInsert	composite_identifier_list_insert	Scientific
InIdListInsertvReg	composite_identifier_list_insert_vreg	Full
InIdListRemove	composite_identifier_list_remove	Scientific
InIdListRemovevReg	composite_identifier_list_remove_vreg	Full
InIdRecordEqualvReg	composite_identifier_record_equal_vreg	Full
InIdRecordGet	composite_identifier_record_get	Scientific
InIdRecordGetvReg	composite_identifier_record_get_vreg	Full
InIdRecordPut	composite_identifier_record_put	Scientific
InIdRecordPutvReg	composite_identifier_record_put_vreg	Full
InImageBarChart	image_bar_chart	Technical
InImageBarChartvReg	image_bar_chart_vreg	Full
InImageConstant	image_constant	Foundation

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InImageConstantVReg	image_constant_vreg	Full
InImageGauss	image_gaussian	Scientific
InImageGaussVReg	image_gaussian_vreg	Full
InImageGreyScale	image_grey_scale	Technical
InImageGreyScaleVReg	image_grey_scale_vreg	Full
InImageRandomNumber	image_random_number	Scientific
InImageRandomNumberVReg	image_random_number_vreg	Full
InImportHist	import_histogram	Foundation
InImportHistVReg	import_histogram_vreg	Full
InImportIIFDataFromPIKS	import_iif_data_from_piks	Foundation - with PIKS only
InImportImage	import_image	Foundation
InImportImageVReg	import_image_vreg	Full
InImportLUT	import_lut	Foundation
InImportLUTVReg	import_lut_vreg	Full
InImportMatrix	import_matrix	Foundation
InImportMatrixVReg	import_matrix_vreg	Full
InImportNbhoodArray	import_neighbourhood_array	Foundation
InImportNbhoodArrayVReg	import_neighbourhood_array_vreg	Full
InImportROIArray	import_roi_array	Technical
InImportROIArrayVReg	import_roi_array_vreg	Full
InImportStaticArray	import_static_array	Technical
InImportStaticArrayVReg	import_static_array_vreg	Full
InImportTuple	import_tuple	Foundation
InImportTupleVReg	import_tuple_vreg	Full
InImportValueBounds	import_value_bounds	Scientific
InImportValueBoundsVReg	import_value_bounds_vreg	Full
InImpulseBoxcar	impulse_boxcar	Scientific
InImpulseBoxcarVReg	impulse_boxcar_vreg	Full
InImpulseDerivGauss	impulse_derivative_of_gaussian	Scientific
InImpulseDerivGaussVReg	impulse_derivative_of_gaussian_vreg	Full
InImpulseDiffGauss	impulse_difference_of_gaussians	Scientific
InImpulseDiffGaussVReg	impulse_difference_of_gaussians_vreg	Full
InImpulseElliptical	impulse_elliptical	Technical
InImpulseEllipticalVReg	impulse_elliptical_vreg	Full
InImpulseGauss	impulse_gaussian	Technical
InImpulseGaussVReg	impulse_gaussian_vreg	Full
InImpulseLaplaceGauss	impulse_laplacian_of_gaussian	Scientific
InImpulseLaplaceGaussVReg	impulse_laplacian_of_gaussian_vreg	Full
InImpulsePyramid	impulse_pyramid	Scientific
InImpulsePyramidVReg	impulse_pyramid_vreg	Full
InImpulseRectangular	impulse_rectangular	Foundation
InImpulseRectangularVReg	impulse_rectangular_vreg	Full
InImpulseSinc	impulse_sinc	Scientific
InImpulseSincVReg	impulse_sinc_vreg	Full
InInputObjectVReg	input_object_vreg	Full - with IIF only
InInsertPixelPlane	insert_pixel_plane	Foundation
InInsertPixelPlaneROI	insert_pixel_plane_roi	Technical

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InInsertPixelPlaneROIVReg	insert_pixel_plane_roi_vreg	Full
InInsertPixelPlaneVReg	insert_pixel_plane_vreg	Full
InLabelObjects	label_objects	Scientific
InLabelObjectsVReg	label_objects_vreg	Full
InLawsTextureFeatures	laws_texture_features	Scientific
InLawsTextureFeaturesVReg	laws_texture_features_vreg	Full
InLevelSlice	level_slice	Scientific
InLevelSliceVReg	level_slice_vreg	Full
InLineProfile	line_profile	Scientific
InLookup	lookup	Foundation
InLookupInterp	lookup_interpolated	Technical
InLookupInterpVReg	lookup_interpolated_vreg	Full
InLookupVReg	lookup_vreg	Full
InLumGeneration	luminance_generation	Technical
InLumGenerationVReg	luminance_generation_vreg	Full
InMarkovSpectrum	markov_spectrum	Scientific
InMarkovSpectrumVReg	markov_spectrum_vreg	Full
InMonadicArith	monadic_arithmetic	Foundation
InMonadicArithVReg	monadic_arithmetic_vreg	Full
InMonadicComplex	monadic_complex	Technical
InMonadicComplexVReg	monadic_complex_vreg	Full
InMonadicLogical	monadic_logical	Foundation
InMonadicLogicalROI	monadic_logical_roi	Technical
InMonadicLogicalROIIVReg	monadic_logical_roi_vreg	Full
InMonadicLogicalVReg	monadic_logical_vreg	Full
InMorphicProcessor	morphic_processor	Foundation
InMorphicProcessorROI	morphic_processor_roi	Technical
InMorphicProcessorROIIVReg	morphic_processor_roi_vreg	Full
InMorphicProcessorVReg	morphic_processor_vreg	Full
InMorphology	morphology	Technical
InMorphologyROI	morphology_roi	Technical
InMorphologyROIIVReg	morphology_roi_vreg	Full
InMorphologyVReg	morphology_vreg	Full
InNbrCount	neighbour_count	Scientific
InNbrCountVReg	neighbour_count_vreg	Full
InNoiseCombination	noise_combination	Scientific
InNoiseCombinationVReg	noise_combination_vreg	Full
InOpenClose	open_close	Scientific
InOpenCloseROI	open_close_roi	Scientific
InOpenCloseROIIVReg	open_close_roi_vreg	Full
InOpenCloseVReg	open_close_vreg	Full
InOpenPort	open_port	Foundation
InOutlierRemoval	outlier_removal	Technical
InOutlierRemovalVReg	outlier_removal_vreg	Full
InOutputObject	output_object	Foundation - with IIF only
InPaintPixels	paint_pixels	Technical
InPaintPixelsVReg	paint_pixels_vreg	Full

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InPolarToCartesian	polar to cartesian	Scientific
InPolarToCartesianVReg	polar to cartesian vreg	Full
InPowerLawScaling	power law scaling	Technical
InPowerLawScalingVReg	power law scaling vreg	Full
InPseudocolour	pseudocolour	Technical
InPseudocolourVReg	pseudocolour vreg	Full
InPutColourPixel	put colour pixel	Foundation
InPutColourPixelVReg	put colour pixel vreg	Full
InPutEntityValue	put iif syntax entity value	Foundation
InPutPixel	put pixel	Foundation
InPutPixelArray	put pixel array	Foundation
InPutPixelArrayROI	put pixel array roi	Technical
InPutPixelArrayROIVReg	put pixel array roi vreg	Full
InPutPixelArrayVReg	put pixel array vreg	Full
InPutPixelRecord	put pixel record	Technical
InPutPixelRecordROI	put pixel record roi	Technical
InPutPixelRecordROIVReg	put pixel record roi vreg	Full
InPutPixelRecordVReg	put pixel record vreg	Full
InPutPixelROI	put pixel roi	Technical
InPutPixelROIVReg	put pixel roi vreg	Full
InPutPixelVReg	put pixel vreg	Full
InReadParseImage	read and parse iif image	Foundation
InRescale	rescale	Foundation
InRescaleROI	rescale roi	Foundation
InRescaleROIVReg	rescale roi vreg	Full
InRescaleVReg	rescale vreg	Full
InResize	resize	Foundation
InResizeROI	resize roi	Foundation
InResizeROIVReg	resize roi vreg	Full
InResizeVReg	resize vreg	Full
InReturnRepositoryId	return repository id	Foundation
InReturnRepositoryIdVReg	return repository id vreg	Full
InROICoordinateArray	roi coordinate array	Technical
InROICoordinateArrayVReg	roi coordinate array vreg	Full
InROIElliptical	roi elliptical	Technical
InROIEllipticalVReg	roi elliptical vreg	Full
InROIPolygon	roi polygon	Technical
InROIPolygonVReg	roi polygon vreg	Full
InROIRectangular	roi rectangular	Foundation
InROIRectangularVReg	roi rectangular vreg	Full
InRotate	rotate	Foundation
InRotateROI	rotate roi	Technical
InRotateROIVReg	rotate roi vreg	Full
InRotateVReg	rotate vreg	Full
InRubberBandScaling	rubber band scaling	Scientific
InRubberBandScalingVReg	rubber band scaling vreg	Full
InSeqAverage	sequence average	Scientific

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InSeqAverageVReg	sequence_average_vreg	Full
InSeqKarhunenLoeve	sequence_karhunen_loeve	Scientific
InSeqKarhunenLoeveVReg	sequence_karhunen_loeve_vreg	Full
InSeqRunningMeasures	sequence_running_measures	Scientific
InSeqRunningMeasuresVReg	sequence_running_measures_vreg	Full
InSetImageAttrs	set_image_attributes	Foundation
InSetImageAttrsVReg	set_image_attributes_vreg	Full
InSplitImage	split_image	Foundation
InSplitImageVReg	split_image_vreg	Full
InSubsample	subsample	Foundation
InSubsampleROI	subsample_roi	Technical
InSubsampleROIIVReg	subsample_roi_vreg	Full
InSubsampleVReg	subsample_vreg	Full
InTemplMatch	template_match	Scientific
InTemplMatchVReg	template_match_vreg	Full
InThreshold	threshold	Foundation
InThresholdVReg	threshold_vreg	Full
InTransformCosine	transform_cosine	Technical
InTransformCosineVReg	transform_cosine_vreg	Full
InTransformFourier	transform_fourier	Technical
InTransformFourierVReg	transform_fourier_vreg	Full
InTransformHadamard	transform_hadamard	Scientific
InTransformHadamardVReg	transform_hadamard_vreg	Full
InTransformHartley	transform_hartley	Scientific
InTransformHartleyVReg	transform_hartley_vreg	Full
InTranslate	translate	Foundation
InTranslateROI	translate_roi	Foundation
InTranslateROIIVReg	translate_roi_vreg	Full
InTranslateVReg	translate_vreg	Full
InUnaryInteger	unary_integer	Foundation
InUnaryIntegerVReg	unary_integer_vreg	Full
InUnaryReal	unary_real	Technical
InUnaryRealVReg	unary_real_vreg	Full
InUnsharpMask	unsharp_mask	Technical
InUnsharpMaskVReg	unsharp_mask_vreg	Full
InValueBounds	value_bounds	Scientific
InValueBoundsVReg	value_bounds_vreg	Full
InVRegAlter	vreg_alter	Full
InVRegAlterVReg	vreg_alter_vreg	Full
InVRegClear	vreg_clear	Full
InVRegClearVReg	vreg_clear_vreg	Full
InVRegCopy	vreg_copy	Full
InVRegCopyVReg	vreg_copy_vreg	Full
InVRegDelete	vreg_delete	Full
InVRegDeleteVReg	vreg_delete_vreg	Full
InVRegSet	vreg_set	Full
InVRegSetVReg	vreg_set_vreg	Full

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
InVRegWait	vreg_wait	Full
InVRegWaitVReg	vreg_wait_vreg	Full
InWallisStatDiff	wallis_statistical_differencing	Technical
InWallisStatDiffVReg	wallis_statistical_differencing_vreg	Full
InWarpControlPoint	warp_control_point	Technical
InWarpControlPointROI	warp_control_point_roi	Technical
InWarpControlPointROIvReg	warp_control_point_roi_vreg	Full
InWarpControlPointVReg	warp_control_point_vreg	Full
InWarpLUT	warp_lut	Technical
InWarpLUTROI	warp_lut_roi	Technical
InWarpLUTROIvReg	warp_lut_roi_vreg	Full
InWarpLUTVReg	warp_lut_vreg	Full
InWarpPolynomial	warp_polynomial	Technical
InWarpPolynomialROI	warp_polynomial_roi	Technical
InWarpPolynomialROIvReg	warp_polynomial_roi_vreg	Full
InWarpPolynomialVReg	warp_polynomial_vreg	Full
InWindowingFunction	windowing_function	Scientific
InWindowingFunctionVReg	windowing_function_vreg	Full
InWindowLevel	window_level	Foundation
InWindowLevelVReg	window_level_vreg	Full
InWindowStatistics	window_statistics	Scientific
InWindowStatisticsVReg	window_statistics_vreg	Full
InZMerge	z_merge	Technical
InZMergeVReg	z_merge_vreg	Full
InZoom	zoom	Foundation
InZoomROI	zoom_roi	Foundation
InZoomROIvReg	zoom_roi_vreg	Full
InZoomVReg	zoom_vreg	Full
In3DSlice	3d_slice	Scientific
In3DSliceVReg	3d_slice_vreg	Full
IrDiffMeasures	difference_measures	Technical
IsnInputObject	input_object	Foundation - with IIF only
IstDecompressEntity	decompress_iif_syntax_entity	Foundation
IstExportImage	export_image	Foundation
IstExportLUT	export_lut	Foundation
IstExportMatrix	export_matrix	Foundation
IstExportNbhoodArray	export_neighbourhood_array	Foundation
IstExportStaticArray	export_static_array	Technical
IstExportTuple	export_tuple	Foundation
IstExportValueBounds	export_value_bounds	Scientific
IstGetEntityValue	get_iif_syntax_entity_value	Foundation
IstGetPixel	get_pixel	Foundation
IstGetPixelArray	get_pixel_array	Foundation
IstGetPixelRecord	get_pixel_record	Technical
ItbaExportROIArray	export_roi_array	Technical
ItbaGetPixelArrayROI	get_pixel_array_roi	Foundation
ItbaGetPixelRecordROI	get_pixel_record_roi	Technical

Table 4 - Function names alphabetical by bound name (continued)

Bound Name	Function Name	Least Profile
ItfSetErrorHandler	set_error_handler	Foundation
ItraAccumulator	accumulator	Foundation
ItraHistShape1D	histogram_shape_1d	Scientific
ItraHistShape2D	histogram_shape_2d	Scientific
ItraLineProfileVReg	line_profile_vreg	Full
ItraShapeMetrics	shape_metrics	Scientific
ItraSpatialMomentsInvariant	spatial_moments_invariant	Scientific
ItraSpatialMomentsScaled	spatial_moments_scaled	Scientific
ItuaExportHist	export_histogram	Foundation
ItuaPerimeterCodeGenerator	perimeter_code_generator	Scientific
ItzPutImageSubrange	put_image_subrange	Foundation
IvAbortAsynchExecution	abort_asynchronous_execution	Scientific
IvCloseGateway	close_iif_gateway	Foundation
IvClosePIKS	close_piks	Foundation
IvClosePIKSEmergency	close_piks_emergency	Foundation
IvClosePort	close_port	Foundation
IvCmpTypeDef	inquire_iif_syntax_component_type_definition	Foundation
IvColourConvMatrix	colour_conversion_matrix	Foundation
IvColourConvMatrixVReg	colour_conversion_matrix_vreg	Full
IvComplexDecomposition	complex_decomposition	Technical
IvComplexDecompositionVReg	complex_decomposition_vreg	Full
IvDeallocateDataObject	deallocate_data_object	Foundation
IvDeallocateDataObjectVReg	deallocate_data_object_vreg	Full
IvDeleteEntity	delete_iif_syntax_entity	Foundation
IvErrorHandler	error_handler	Foundation
IvErrorLogger	error_logger	Foundation
IvExtrema	extrema	Foundation
IvGetColourPixel	get_colour_pixel	Foundation
IvGetColourPixelVReg	get_colour_pixel_vreg	Full
IvInquireChainEnvironment	inquire_chain_environment	Full
IvInquireChainStatus	inquire_chain_status	Full
IvInquireElements	inquire_elements	Foundation
IvInquireEntityChc	inquire_iif_syntax_entity_choice	Foundation
IvInquireEntityNumber	inquire_iif_syntax_entity_number	Foundation
IvInquireEntityOptional	inquire_iif_syntax_entity_optional	Foundation
IvInquireEntityTypeDef	inquire_iif_syntax_entity_type_definition	Foundation
IvInquireEntityValidity	inquire_iif_syntax_entity_validity	Foundation
IvInquireFunctionality	inquire_supported_functionality	Foundation
IvInquireGatewayStatus	inquire_iif_gateway_status	Foundation
IvInquireImage	inquire_image	Foundation
IvInquireIndexAssignment	inquire_index_assignment	Technical
IvInquireNonImageObject	inquire_non_image_object	Foundation
IvInquirePIKSImp	inquire_piks_implementation	Foundation
IvInquirePIKSStatus	inquire_piks_status	Foundation
IvInquirePort	inquire_port	Foundation
IvInquireRepository	inquire_repository	Foundation
IvInquireResampling	inquire_resampling	Foundation

Table 4 - Function names alphabetical by bound name (concluded)

Bound Name	Function Name	Least Profile
IvMoments	moments	Foundation
IvMomentsVReg	moments_vreg	Full
IvOpenGateway	open_iif_gateway	Foundation
IvOpenPIKS	open_piks	Foundation
IvReleaseEntityId	release_iif_syntax_entity_identifier	Foundation
IvResetGateway	reset_iif_gateway	Foundation
IvResetPort	reset_port	Foundation
IvSetGlobals	set_globals	Foundation
IvSetGlobalsVReg	set_globals_vreg	Full
IvSetIndexAssignment	set_index_assignment	Technical
IvSetIndexAssignmentVReg	set_index_assignment_vreg	Full
IvSynch	synchronize	Scientific
IvVRegConditional	vreg_conditional	Full
IvVRegConditionalVReg	vreg_conditional_vreg	Full
IvVRegGet	vreg_get	Full

5 Data type definitions

5.1 Mapping of data types

The architecture of the Image Processing and Interchange Standard, ISO/IEC 12087-1, defines sets of abstract data types. This clause gives the mapping from those data types to the data defined later in this part of 12088.

The numerous types defined in IPI Parts 1, 2 and 3 and the association with the application data are bound as identified in Table 5. IPI-PIKS identifies internal data types which are implementation-dependent and may not be of the same type as the external representation of related data types. Most parameter and application data types are common to both IPI-IIF and IPI-PIKS. The precision of application data is specified by the application to the implementation. The precision of internal data is given by ISO/IEC 12087 to require at least 8 bits for non-negative integers, 16 bits for signed integers, 32 bits for real values and 32 bits for each component, real and imaginary, of complex value representations.

Table 5 - Data types

Data Type	IPI-PIKS Internal		Parameters		External Image	
	Functional	C Binding	Functional	C Binding	Functional	C Binding
null	NULL	NULL	NULL	NULL		
Boolean	BD		BP	Ipbool	BI	Imbool
enumerated			EP	enum	enumerated	enum
non-negative integer	ND		NP	Ipoint	NI	Imuint
signed integer	SD		SP	Ipint	SI	Imint
fixed point integer					TI	Imfixed
real arithmetic	RD		RP	Ipfloat (PIKS) Ipdouble (IIF)	RF	Imfloat
complex arithmetic	CD		CP	Ipscomplex	CF	Imcomplex
character string	CS		CS	char		
data object identifier	ID	Idnxxx	IP	Ipnxxx		

Real and complex IPI-PIKS internal data types are not required in an implementation of the foundation profile of IPI-PIKS.

The only IPI-PIKS internal data types represented in this binding are NULL, identifiers available to the application for referencing objects and identifiers returned to the application for assignment of objects to functional elements. Other internal IPI-PIKS data types are left as implementation dependent. Data object identifiers, designated Idnxxx and Ipnxxx, tabulated above, are assigned pertinent type names defined in 5.2.3.

5.2 Environment type definitions

The data types defined in this section allow for the ease of porting application programs between different environments. These types are used as the basis for other data types declared in this part of ISO/IEC 12088.

5.2.1 External physical image data types

Each of the external physical image data types shall be defined for pixel representation by the implementation as a specific type suitable for containing their appropriate values used by the implementation. Suggest:

```
typedef      unsigned char    Imbool;          /* BI data type          */
typedef      enum {names}     Imenum_name;     /* enumerated data type   */
typedef      unsigned char    Imuint;          /* NI data type          */
typedef      int               Imint;           /* SI data type          */
typedef      int               Imfixed;         /* TI representation of real data type */
typedef      float             Imfloat;         /* RF float representation of real data type */

typedef struct {
    Imfloat      xReal;           /* real part              */
    Imfloat      xImaginary;     /* imaginary part         */
} Imscomplex;    /* CF data type          */
```

It is suggested that the Boolean external physical image data type, BI, be represented in bitmap form, one pixel per bit, eight pixels per byte. As optioned to import/export utilities, pixels may be ordered from the most significant bit or least significant bit of the image data bytes.

The enumerated pixel data type is specific to the IPI-IIF. An implementation shall provide individually named (Imenum_name) declarations, each with a list of unique enumerated names, for each collection of enumerated pixel types supported.

The fixed point integer data type is unique to IPI-PIKS. This can be represented by the signed integer data type in IPI-IIF.

The real and complex data types are not required in an implementation of the foundation profile of IPI-PIKS.

5.2.2 Basic parameter data types

5.2.2.1 IPI-PIKS parameter data types

Each of these data types shall be defined for an IPI-PIKS implementation as a specific type suitable for containing their appropriate values used by the implementation. Suggest:

```
typedef      unsigned char    Ipbool;          /* BP data type          */
typedef      unsigned long int Ipoint;         /* NP data type          */
typedef      int              Ipoint;          /* SP data type          */
typedef      float            Ipfloor;         /* RP real data type     */

typedef struct {
    Ipfloor      xReal;          /* real part             */
    Ipfloor      xImaginary;     /* imaginary part        */
} Ipscomplex; /* CP data type          */
```

5.2.2.2 IPI-IIF parameter data types

Each of these data types shall be defined for an IPI-IIF implementation as a specific type suitable for containing their appropriate values used by the implementation. Suggest:

```
typedef      unsigned char    Ipbool;          /* BP data type          */
typedef      unsigned long int Ipoint;         /* NP data type          */
typedef      int              Ipoint;          /* SP data type          */
typedef      double           Ipdouble;        /* RP real data type     */

typedef struct {
    Ipdouble      xReal;          /* real part             */
    Ipdouble      xImaginary;     /* imaginary part        */
} Ipscomplex; /* CP data type          */
```

5.2.3 Data object identifiers

5.2.3.1 IPI-PIKS object identifiers

Each of these data types shall be defined by the implementation as a type suitable for containing their appropriate representations used by the implementation. Suggest:

typedef	void	*Idnchain;	/* IPI-PIKS internal chain object	*/
typedef	void	*Idnid_array;	/* IPI-PIKS internal image id array	*/
typedef	void	*Idnid_list;	/* IPI-PIKS internal image id list	*/
typedef	void	*Idnid_record;	/* IPI-PIKS internal image id record	*/
typedef	void	*Idnhist;	/* IPI-PIKS internal histogram arrays	*/
typedef	void	*Idnimage;	/* IPI-PIKS internal image objects	*/
typedef	void	*Idnlut;	/* IPI-PIKS internal lookup tables	*/
typedef	void	*Idnmatrix;	/* IPI-PIKS internal matrix objects	*/
typedef	void	*Idnnbhood;	/* IPI-PIKS internal neighbourhood arrays	*/
typedef	void	*Idnpixel_record;	/* IPI-PIKS internal pixel record	*/
typedef	void	*Idnrepository;	/* IPI-PIKS repository object	*/
typedef	void	*Idnroi;	/* IPI-PIKS internal region of interest objects	*/
typedef	void	*Idnstatic;	/* IPI-PIKS internal static arrays	*/
typedef	void	*Idntuple;	/* IPI-PIKS internal tuples	*/
typedef	void	*Idnbounds;	/* IPI-PIKS internal value bounds lists	*/
typedef	void	*Idnvreg;	/* IPI-PIKS internal virtual register objects	*/
typedef	void	*Ipnarg;	/* external function reloadable arguments	*/
typedef	FILE	*Ipnerror;	/* external error file	*/
typedef	void	*Ipnport;	/* external input/output port	*/
typedef	void (Ipint Ipint Idnchain Ipnerror Ipesynchronicity_state	(*Iptferror_function) iCode, iElement, nChain, nError, eSynchronicityState);	/* error handler function pointer	*/