

ICS — Common Metadata for Document Production Workflows

Abstract

This CIP4 JDF Interoperability Conformance Specification (ICS) defines a PDL independent architecture for encoding standardized metadata embedded in structured PDL data. Such metadata is useful to receiving systems for determining the production intent of the PDL pages and to aid the creation of parameterized JDF-based job tickets.



CIP4 THANKS ITS PARTNER LEVEL MEMBERS



Copyright notice

Copyright © 2000-2010, International Cooperation for the Integration of Processes in Prepress, Press and Postpress hereinafter referred to as CIP4. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Specification and associated documentation files (the "Specification") to deal in the Specification, including without limitation the rights to use, copy, publish, distribute, and/or sublicense copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the following conditions. The above copyright notice and this permission notice must be included in all copies or substantial portions of the Specification.

THE SPECIFICATION IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED, OR OTHERWISE, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT WILL CIP4 BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SPECIFICATION OR THE USE OR OTHER DEALINGS IN THE SPECIFICATION.

Except as contained in this notice or as allowed by membership in CIP4, the name of CIP4 must not be used in advertising or otherwise to promote the use or other dealings in this Specification without prior written authorization from CIP4.

Licenses and Trademarks

International Cooperation for Integration of Processes in Prepress, Press and Postpress, CIP4, Job Description Format, JDF and the CIP4 logo are trademarks of CIP4.

Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Contents

Page

Foreword	iv
Introduction.....	v
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions	2
4 Notations	3
5 Conformance	3
6 Technical Requirements.....	3
6.1 Common Metadata Hierarchy.....	3
6.2 Metadata keys.....	4
6.3 Metadata values.....	4
6.4 Document part hierarchy.....	4
6.5 Defining metadata within a document part.....	5
6.5.1 Encoding metadata keys	5
6.5.2 Encoding metadata values	6
7 The CIP4 Common Metadata Hierarchy	6
7.1 The Metadata level.....	6
7.2 The Summary level.....	7
7.3 The Recipient level	8
7.4 The Production level	8
8 Common metadata structures	10
8.1 Contact Information	10
Annex A (informative) Common Metadata Hierarchy.....	12
A.1 Registered Second Class Name Prefixes	12
Annex B (informative) Metadata Encoding Examples.....	13
B.1 XML metadata encoding	13
B.2 PDF metadata encoding.....	15

Foreword

The International Cooperation for the Integration of Processes in Prepress, Press and Postpress Organization (CIP4) is a not-for-profit association and is registered with the U.S. Department of Justice and the Federal Trade Commission as a Standards Development Organization. CIP4 is established in Switzerland, it has no regular offices; rather is a global organization with representatives in several countries. CIP4 brings together vendors, consultants and end-users in the print communications, Graphic Arts industry, and associated sectors, covering a variety of equipment, software, peripherals, and processes. Members participate in focused working groups to define future versions of Job Definition Format (JDF), to study user requirements, and to design a JDF Software Development Kit (SDK).

Working Groups, such as "Prepress" or "Digital Print Workflow", discuss and agree to changes that are pertinent to their area. These changes are submitted to a Technical Steering Committee (TSC) that is responsible for reviewing and approving all changes to the specifications. The TSC is charged with settling issues that arise between Working Groups and ensuring that the overall architecture of the JDF specification and schema remain sound.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CIP4 shall not be held responsible for identifying any or all such patent rights.

This document was prepared by the Document Metadata sub-group of the Digital Print Workflow Working Group.

Introduction

This document defines an Interoperability Conformance Specification for the definition of PDL-independent standard metadata keys and their meanings for the purposes of aiding the creation of parameterized JDF-based job tickets. The published metadata keys are intended to not only be useful to the CIP4 community, but also for users of non-JDF based job tickets.

The intent is to accomplish this through standardizing the use of metadata in content creation. Note that product or service ICS documents will have the responsibility of specifying which metadata keys are required to be supported or prohibited.

PDL files that represent content pages do not normally contain information identifying the purpose of these content pages. Standardized metadata is a simple mechanism that allows for the exchange of information regarding these content pages to aid the receiver of the PDL files in determining the intended use of those content pages in the final print product. By understanding the intended use of content pages the receiver of the PDL file can make more informed decisions regarding the production process for the final print product.

This initial version focuses on defining a PDL independent architecture for encoding standardized metadata into PDL files. This specification defines rules for encoding metadata that relies on and adheres to the PDL specified method of encoding metadata, where available. The informative annexes of this document describe known encodings for some PDLs (e.g. ISO 16612-2 (PDF/VT), ISO 32000-1 (PDF), PPML, ANSI CGATS.20:2002 (PPML/VDX) and ISO 16612-1 (PPML/VDX).

This version of the common metadata specification defines standardized metadata to:

- provide summary information to aid in optimizing or simplification of the production process
- provide contact information of the owner of the PDL files
- identify what content pages are intended to represent (e.g. a brochure, letter, postcard, etc)
- identify the intended recipient of each of the content pages for variable document printing applications

Future versions of this specification are expected to be published that standardize print application specific metadata using the architecture defined in this specification.

ICS — Common Metadata for Document Production Workflows

1 Scope

This document is an Interoperability Conformance Specification (ICS) for the definition of PDL-independent standard metadata keys and their meanings for the purposes of aiding the creation of parameterized JDF-based job tickets. The metadata keys defined by this ICS are intended to not only be useful to the CIP4 community, but also for users of non-JDF based job tickets.

PDL specific metadata encodings are not defined by this ICS. The informative annexes of this document provide examples of known encodings for specific non-proprietary PDL formats.

This ICS defines a base conformance level that includes the syntax and semantics of metadata keys for common use cases in document production. This base conformance level does not require support for any specific set of metadata keys by a conforming reader or conforming writer. CIP4 Service ICS documents will standardize the set of metadata keys that a conforming reader or conforming writer must support. CIP4 welcomes any proposals for new standardized keys to be added to the CIP4 metadata hierarchy in future versions of this specification.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Adobe PDF Reference, fifth edition, version 1.6., Adobe Systems Incorporated (ISBN 0-321-30474-8). Available from internet <http://www.npes.org/standards/toolspdfx.html>

Adobe PostScript Reference, third edition, Adobe Systems Incorporated (ISBN 0-201-37922-8).

ISO 8601:2004 *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO 3166-1:2006 *Codes for the representation of names of countries and their subdivisions - Part 1: Country codes*

ISO 16612-2 *Graphic technology — Variable data exchange — Part 2: Using PDF/X-4 and PDF/X-5 (PDF/VT-1 and PDF/VT-2)*

Extensible Markup Language (XML) 1.0 (Second Edition), 6 October 2000, World Wide Web Consortium, Available from internet <<http://www.w3.org>>

JDF Specification, Release 1.4, 2008, CIP4 Organization, Available from internet <<http://www.CIP4.org>>

XML Path Language (XPath) 1.0 specification, 16 November 1999. Available from internet <<http://www.w3.org/tr/xpath>>

XMP Specification, June 2005, Adobe Systems Incorporated. Available from internet <<http://www.npes.org/standards/toolspdfx.html>>

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply:

3.1

document

collection of related document parts

3.2

document part

set of related pages and/or related sets of pages

EXAMPLE chapter pages of a book or all sets of pages intended for a recipient

3.3

document part hierarchy

hierarchical data structure that specifies the organization of document parts

3.4

document part metadata

metadata associated with a document part

3.5

JDF

Job Definition Format

3.6

PDL

Page Description Language

EXAMPLE PostScript, PDF and PCL .

3.7

job definition

information that specifies the production requirements and workflow of a unit of work involving purposing PDL content to one or more messaging channels

3.8

job ticket

electronic specification of process control for print production

3.9

print product

outcome of the processing of a document through a print manufacturing process

EXAMPLE a perfect bound book or postcard.

3.10

product part

part of a print product

EXAMPLE the cover part of a saddle-stitched booklet.

3.11

recipient

the person or institution that receives a print product

3.12**conforming reader**

software application that is able to read and process PDL data in accordance with this ICS

3.13**conforming writer**

software application that is able to write PDL data in accordance with this ICS

3.14**ICS**

Interoperability Conformance Specification

3.15**reused content**

part of the PDL data that is included in more than one page definition by reference

4 Notations

The names of keys, PDL keywords and other predefined names are written in a bold sans serif type font; for example, the key **Trapped**.

Operands of operators or values of keys are written in an italic sans serif font; for example, the *False* value for the **Trapped** key.

5 Conformance

This ICS defines a base conformance level for the exchange of metadata for the purposes of aiding the creation of parameterized job tickets. The Base Conformance Level defines the syntax and semantics of common metadata properties.

Conforming PDL data shall conform to all the technical requirements set out in Clauses 6 and 7 of this ICS. Conforming PDL data shall include the **CIP4/Metadata/Conformance**, **CIP4/Metadata/Creator** and **CIP4/Metadata/ModificationDate** properties at the root of the document part hierarchy of the PDL data as defined in Clause 7 of this ICS.

A conforming writer is a software application that shall write PDL data conforming to the format specification of that PDL format and contains metadata conforming to the requirements defined in this ICS. The metadata of the PDL data conforming to this ICS shall be encoded in accordance with the metadata encoding requirements of the PDL specification.

A conforming reader is a software application that shall read and appropriately process PDL data conforming to a PDL format specification and shall read and appropriately process the metadata encoded in that PDL data conforming to the requirements defined in this ICS.

6 Technical Requirements**6.1 Common Metadata Hierarchy**

This ICS defines metadata properties with key names that are chosen from the common metadata hierarchy.

The first level of the common metadata hierarchy shall consist of second class name prefixes as defined in *Adobe PDF Reference*, Appendix E. The registrant of the second class name prefix controls the hierarchy

CIP4

under that second class name prefix. CIP4 has registered the *CIP4* second class name prefix which shall be used for all metadata properties defined in this ICS and shall only be used by CIP4 specifications.

Each leaf node of the common metadata hierarchy shall be the name of an individual metadata property. This name shall conform to the rules of the XML name token and shall not contain colon or forward slash characters.

The name of each level in the common metadata hierarchy shall conform to the rules for the XML name token and shall not contain colon or forward slash characters.

NOTE 1 The requirement to restrict names in the common metadata hierarchy to adhere to the XML name token facilitates the encoding of metadata in XML formats. PDF and non-XML based formats are in general more liberal in their naming requirements.

NOTE 2 The requirement for key names to be an XML name token implies that all keys are represented internally as unicode characters as defined in the XML specification.

A vendor wishing to add private metadata properties and levels into the CIP4 hierarchy may do so but shall explicitly identify those private metadata properties and levels by specifying an alternate second class name prefix for that property or level. An alternate second class name prefix shall be specified by prefixing the name of a metadata property or level with a second class name prefix followed by a colon character.

EXAMPLE A vendor that is using the second class name prefix **ACME** that wishes to encode a value for a key named **foobar** in the **CIP4/Recipient** hierarchy will therefore use a metadata property called **CIP4/Recipient/ACME:foobar**.

NOTE 3 A vendor wishing to add private metadata properties is required to register and use a second class name prefix for that private metadata. This ensures maximum interoperability as the CIP4 defined hierarchy can only contain known standard keys with a predefined meaning. The private keys are therefore clearly separated and allow the reader to skip them more readily.

6.2 Metadata keys

Each metadata property shall have a second class name prefix. The second class name prefix of a metadata property shall be the name of the top-most level of the common metadata hierarchy in which this metadata property is defined unless the metadata property explicitly specifies a different second class name prefix (using the colon syntax).

Each metadata property shall be identified by a key. The name of that key shall be constructed from the names of each level in the common metadata hierarchy that the metadata property is part of, separated by forward slashes.

Each metadata property shall have a context. The name of that context shall be constructed from the names of each of the non-leaf levels of the common metadata hierarchy that the metadata property is part of, separated by forward slashes.

Each metadata property shall have a name. The name of a metadata property shall be the name of the level without any second class name prefix.

6.3 Metadata values

The value of each metadata property, or any part of that value, shall only be of type boolean, numeric (integer & real), string, name, array or dictionary as defined in *Adobe PDF Reference*, 3.2. Dictionary keys shall adhere to the rules for an XML name token.

6.4 Document part hierarchy

A document part hierarchy is a tree of document parts where the leaf nodes represent one or more pages. Typically each node in the document part hierarchy can have metadata associated with that document part.

EXAMPLE 1 PPML defines a fixed document part hierarchy consisting of the **PPML**, **JOB**, **DOCUMENT** and **PAGE** levels. Each of these levels may contain one or more **METADATA** elements to attach metadata to that level.

EXAMPLE 2 ISO 16612-2 (PDF/VT) and ISO 32000-2 define a variable depth document part hierarchy. Each **DPart** dictionary represents a document part. Each **DPart** dictionary can have a **DPM** key whose value is a dictionary representing the metadata.

EXAMPLE 3 The Document Structuring Conventions (DSC) for PostScript allows PostScript comments to be used to attach arbitrary information to a single page or to all of the pages.

PDL data for which the PDL format specification defines a document part hierarchy may contain metadata properties defined in this ICS. Unless the PDL format specification specifies otherwise, metadata properties defined for a given document part shall not be considered to apply to any document parts that are child nodes of that document part.

6.5 Defining metadata within a document part

Each document part node of a document part hierarchy may have one or more metadata properties from the common metadata hierarchy specified within it. The key and value of each metadata property shall be encoded in accordance with the requirements of the PDL specification and the requirements set forth in Sub-clauses 6.5.1 and 6.5.2.

6.5.1 Encoding metadata keys

The metadata hierarchy shall be encoded as a collection of dictionaries, where each dictionary represents a level in the hierarchy. Each such dictionary contains a key identifying the name of a sub-level or a key identifying a metadata property.

Each top-most hierarchy level shall be encoded as a PDL metadata property whose name is *Root* and shall use the second class name prefix of the top-most hierarchy level. The value of that PDL metadata shall be a dictionary value representing the hierarchy contained within that top-most hierarchy level.

EXAMPLE All the metadata properties defined in this specification are contained in a PDL metadata property with name **CIP4:Root**.

Each level in the hierarchy shall be encoded as a key within the dictionary value associated with the parent level of that level. That key shall equal the name of that level. The value for that key shall be a dictionary value representing the hierarchy contained within that level.

Each metadata property shall be encoded as a key within the dictionary value associated with the hierarchy level in which that metadata property is defined. The name of that key shall equal the name of the metadata property. The value of that key shall be encoded in accordance with section 6.5.2.

If the PDL format specification is using XML elements or attribute names to encode the name of a PDL metadata property and the PDL format specification allows the use of XML namespaces then each XML element or attribute shall use a namespace consisting of the second class name prefix of the metadata property or level the XML element encodes, prefixed by *urn:cip4.org:CommonMetadata:*. For the metadata properties and levels in this specification the namespace *urn:cip4.org:CommonMetadata:CIP4* shall be used. The XML element or attribute name shall be equal to the name of the metadata property or level without the second class name prefix.

If the PDL specification is using PDF or PostScript syntax to encode the name of a PDL metadata property then a name object shall be used whose name consists of the second class name prefix of the metadata property followed by an underscore symbol and the name of the metadata property.

The PDL specification may require certain characters to be specially encoded, such as # notation, XML character escaping, UTF-8 or UTF-16 encoding. A conforming writer shall apply all such required encodings before writing the name of the key. A conforming reader shall reverse all these required encodings to obtain the original name of that key.

6.5.2 Encoding metadata values

If a PDL specification may require the value of a PDL metadata property to be encoded as content for an XML element and the PDL specification allows the use of additional namespaces then the value of that PDL metadata property shall be encoded in accordance with the following rules:

- A string value shall be represented as the sequence of valid XML characters.
- A name value shall be represented as the sequence of valid XML characters that form the string representation of the name value (i.e. without the leading slash).
- An array value shall be represented as a sequence of XML child elements, and each of the XML child elements shall have the XML element name **Item**. Each such representative XML child element shall have a value derived from the corresponding array element entry according to these rules for encoding values for metadata properties.
- A dictionary value shall be represented as a sequence of XML child elements. Each XML child element shall have an element name that is the same as the name of the key associated with the corresponding dictionary entry. Each XML child element shall have a value derived from the key value according to these rules for encoding values for metadata properties.
- Boolean and numeric values shall be represented as the shortest sequence of valid XML characters exactly representing the value using PDF syntax.

If the PDL specification does not allow for the use of additional namespaces then the value of the metadata property shall be encoded as a string value instead.

NOTE The above rule implies that the content will be encoded as flat text without any hierarchy.

A PDL specification may require the value of a metadata property to be encoded using PDF or PostScript syntax. If this is the case, the value of that metadata property shall be encoded using the syntax defined in *Adobe PDF Reference*. Each dictionary key name shall be prefixed by the second class name prefix of the metadata property or level that defines that dictionary key name followed by the underscore character. Dictionary key values shall use UTF-8 encoding. Those key values may use # notation as defined in *Adobe PDF Reference*, Appendix E. Examples of PDF based metadata encoding can be found in B.2.

If a PDL specification requires the value of a PDL metadata property to be encoded as a string then the value of the metadata property shall be encoded into a string using the above rules for encoding the value as content for an XML element. If the PDL specification allows those string values to use UTF-8 or UTF-16 encodings then that string value shall be encoded using one of those encodings. If such an encoding is not supported then all non-ASCII characters shall use XML character escaping.

The PDL specification may require certain characters to be specially encoded, such as # notation, XML character escaping, UTF-8 or UTF-16 encoding. A conforming writer shall apply all such required encodings before writing the value of a metadata property. A conforming reader shall reverse all these required encodings to obtain the original value of that metadata property.

7 The CIP4 Common Metadata Hierarchy

7.1 The Metadata level

The **CIP4/Metadata** level contains metadata properties that shall only be used in the root of the document part hierarchy. This level provides information regarding the PDL data as a whole.

The **CIP4/Metadata/Conformance** property shall have a value of type string that indicates the list of ICS's separated by spaces to which all the metadata in the PDL data adheres. A value of *base* may be used if no other more restrictive ICS applies. Each ICS that restricts the use of metadata properties defined in this ICS should include a required value for this metadata property that uniquely identifies that ICS. That required value shall adhere to the requirements for XML name token.

The **CIP4/Metadata/Creator** property shall have a value of type string that identifies the conforming writer of that metadata.

The **CIP4/Metadata/ModificationDate** property shall have a value of type string that identifies the date at which the PDL data was last modified or created. The data shall conform to the format specified in ISO 8601:2004.

NOTE 1 The above property allows detection of the modifications to PDL data by a non-conforming writer. If the PDL specification already encodes a mandatory last modification date, the OS modification date by itself is not necessarily sufficient to detect such modifications.

The **CIP4/Metadata/JobID** property shall have a value of type string that identifies the job or contract to which the PDL data as a whole belongs.

The **CIP4/Metadata/Accounting** sub-level shall identify the contact information of where to send the bill for the production of the PDL data. The properties of the **CIP4/Metadata/Accounting** sub-level are defined in 8.1.

The **CIP4/Metadata/Sender** sub-level shall identify the contact information for the sender or originator of the PDL data. The properties of the **CIP4/Metadata/Sender** sub-level are defined in 8.1.

The **CIP4/Metadata/Author** sub-level shall identify the contact information for the author of the PDL data. The properties of the **CIP4/Metadata/Author** sub-level are defined in 8.1.

The **CIP4/Metadata/Administrator** sub-level shall identify the contact information regarding the execution of the PDL data. The properties of the **CIP4/Metadata/Administrator** sub-level are defined in 8.1.

The **CIP4/Metadata/Managed** property shall have a value of type array of name that identifies the names of the metadata properties that may need to be updated whenever the PDL data is modified by an editor functioning as both a conforming reader and writer.

The **CIP4/Metadata/NotUpdated** property shall have a value of type array of name that identifies the names of the metadata properties listed in **CIP4/Metadata/Managed** that were not updated when the PDL data was modified.

NOTE 2 Typically keys are listed in the value of the above property because that were not supported or understood by the editor of the PDL data.

7.2 The Summary level

The **CIP4/Summary** level shall contain metadata properties that have values that can be determined by inspecting the PDL data.

The **CIP4/Summary/PageCount** property shall have a value of type numeric. The value of this property shall equal the total number of pages in the document part in which this property is specified.

The **CIP4/Summary/Uniform** property shall have a value of type dictionary. The value of each dictionary key shall be of type boolean. The following keys have been defined:

- *Orientation*: this value shall only be used for a document part if and only if all the pages of that document part have the same orientation (e.g. all portrait or all landscape).
- *Size*: this value shall only be used for a document part if and only if all the pages of that document part have the same size independent of orientation (e.g. an A4 landscape page is the same size as an A4 portrait page).
- *Color*: this value shall only be used if and only if all of the page content of the document part uses multiple colorants.
- *Monochrome*: this value shall only be used if and only if all the graphical content of all pages of the document part use a single colorant other than black.

CIP4

- *Black*: this value shall only be used if and only if all the graphical content of all pages of the document part use black colorant.

NOTE When both *Size* and *Orientation* are true the dimensions of all pages are the same.

The **CIP4/Summary/UniformRecipientStructure** property shall have a value of type boolean. This property shall only be used in the root of the document part hierarchy and shall only have the value *true* if and only if the structure of each document part intended for a single recipient has the same structure and corresponding document part leaf nodes have the same number of pages.

The **CIP4/Summary/UniformNodeStructure** property shall have a value of type boolean. This property shall only have the value *true* if and only if the structure of each document part contained in the document part in which this property is specified, has the same structure and the corresponding document part leaf nodes have the same number of pages.

NOTE 1 The *UniformNodeStructure* value can only be used on non-leaf document part nodes.

The **CIP4/Summary/RecipientCount** property shall have a value of type numeric. The value of this property shall indicate the total number of recipients for which there are document parts contained within the document part in which this property is specified.

The **CIP4/Summary/Content/Referenced** property shall have a value of type array of string. The value of this property shall contain the identifier for each reused content definition referenced from pages contained within the document part in which this property is specified. For PPML the value of the OCCURRENCE/@*Name* shall be used. For ISO 16612-2 (PDF/VT), the value of the */GTS_XID* key of the recurring **XObject** shall be used. For PostScript the name of the recurring form resource shall be used.

The **CIP4/Summary/Content/ReferenceCount** property shall have a value of type array of integer. Each entry in this array value shall be the number of references, within the document part in which this property is defined, for the reused content indicated in the corresponding entry of the **CIP4/Summary/Content/Referenced** property on the same document part.

7.3 The Recipient level

The **CIP4/Recipient** level contains metadata properties with information regarding the intended recipient of the pages in a given document part.

The **CIP4/Recipient/Uniqueld** property shall have a value of type string. The value of this property shall uniquely identify the recipient within the PDL data. For any given page in the PDL data there shall only be a single document part that specifies this property.

The **CIP4/Recipient/Contact** sub-level shall define the contact information for the recipient linked to the document part on which properties in the **CIP4/Recipient/Contact** level are defined. The properties of the **CIP4/Recipient/Contact** sub-level are defined in 8.1. These properties shall only be present on document parts that also specify the **CIP4/Recipient/Uniqueld** property.

7.4 The Production level

The **CIP4/Production** level contains metadata properties that may be used to parameterize the job ticket.

The **CIP4/Production/CopyCount** property shall have a value of type integer and shall be a positive value. The value of this property shall indicate the number of copies requested of a document part.

NOTE 1 In the case where a contained document part also specifies the **CIP4/Production/CopyCount** property then the total number of copies are multiplied.

The **CIP4/Production/Part/Class** property shall have a value of type string restricted to the XML NMTOKEN values. The value of this property shall indicate a class of production requirements to be applied to the document part on which this property is defined.

NOTE 2 Whenever possible the CIP4/Production/Part/ProductType should be used to drive production requirements instead of the above CIP4/Production/Part/Class.

The **CIP4/Production/Part/ProductType** property shall have a value of type name. The value of this property shall indicate what the document part represents. The name shall be one of the following:

- *Map*
- *Envelope*
- *Label*
- *Box*
- *Poster*
- *Postcard*
- *Newspaper*
- *Chapter*
- *Section*
- *Letter*
- *Body*: Generic content inside of a Cover.
- *Book*: Body with a Cover and a Spine
- *Booklet*: Body with a Cover without a Spine (typically stapled)
- *Brochure*: A single folded sheet
- *Leaflet*: A single unfolded sheet
- *BusinessCard*
- *Cover*: A single sheet covering a side of a print product.
- *WrapAroundCover*: A single sheet containing the Front Cover, Spine and Back Cover.
- *Spine*: The binding side of a print product
- *Insert*: A product part intended to be inserted into a print product
- *Jacket*: Hard cover case jacket
- *CoverLetter*: A letter accompanying another print product
- *ResponseCard*: A SelfMailer to respond to an offer
- *SelfMailer*: A document to be sent via the post without an additional envelope.
- *Other*: **none of the above.**

EXAMPLE A PostCard is a single-sheet self-mailer. A wafer-sealed document is also a self-mailer.

CIP4

The **CIP4/Production/Part/Side** property shall have a value of type name. The name shall be one of the following:

- *Front*: to be used for the front side of a product part or for the outside of a cover
- *Back*: to be used for the back side of a product part or for the inside of a cover
- *Both*: to be used for the combined front & back side of a product part

The **CIP4/Production/Part/BindingSide** property shall have a value of type name. The name shall be one of the following:

- *Left*: to be used for binding on the left edge of the product part
- *Right*: to be used for binding on the right edge of the product part
- *Top*: to be used for binding on the top edge of the product part
- *Bottom*: to be used for binding on the bottom edge of the product part

8 Common metadata structures

8.1 Contact Information

Contact information is encoded with properties relative to a level of the metadata hierarchy.

The **Person** property shall have a value of type dictionary or type string. The dictionary value may contain the following keys, which each shall have a value of type string:

- *NamePrefix*: a prefix to the name of the recipient., e.g. *Mr.*, *Ms.*, *Dr.*, etc.
- *NameSuffix*: a suffix to the name of the recipient, e.g. *jr.*, *III*, etc.
- *FirstName*: the first name of the recipient
- *AdditionalNames*: the middle name(s) of the recipient
- *LastName*: the last name of the recipient
- *JobTitle*: the job function of the recipient within the organization
- *Organization*: the name of the organization to which the recipient belongs
- *Department*: the name of the department within the organization to which the recipient belongs

NOTE A string value is used where the original database does not provide all the individual details of the persons name.

The **Address** property shall have a value of type dictionary. The value of this property shall identify the postal address of the recipient within the PDL data. The dictionary value may contain the following keys, which each shall have a value of type string unless otherwise stated:

- *AddressLines*: an array of string representing the complete address as a sequence of address lines
- *Block*: the name or number of the block
- *Level*: the level within the building

- *Suite*: the name or number of the suite within the building
- *StreetName*: the name or number of the street where the building is located
- *CivicNumber*: the civic number of the building
- *Street*: contains the complete street address which is a combination of the values of the *Block*, *Level*, *Suite*, *StreetName* and *CivicNumber* when they are not available separately
- *City*: the name of the city in which the building or postal box is located
- *PostalCode*: the postal code of the building (may include the routing code)
- *PostBox*: the postal box number
- *Region*: the name of the region (e.g. state, province, etc.) in which the building or postal box is located
- *Country*: the name of the country in which the building or postal box is located
- *CountryCode*: the ISO 3166-1 code of the country in which the building or postal box is located

NOTE This ICS does not define the method to convert the above dictionary values into address lines as this is country and language specific.

NOTE An address may be encoded as a set of address lines, metadata or both. The latter case may occur where the original database does not provide all the individual details of the address.

The **ComChannel** property shall have a value of type array of dictionary. The value of this property shall identify means of contacting the recipient within the PDL data. Each dictionary value may contain the following keys:

- **ChannelType**: name value indicating the type of the communication channel to use to contact the recipient. One of the following name values shall be used: *Phone*, *Email*, *Fax* or *InstantMessaging*.
- **ChannelTypeDetails**: string value with details regarding the communication channel. For **Phone** channels one of the following values shall be used: *LandLine*, *Mobile*, *Secure* or *ISDN*. For *InstantMessaging* channels this value shall define the name of the service vendor.
- **ChannelUsage**: array of name values indicating the purpose of the communication channel. Each array entry shall use one of the following name values: *Business*, *Private*, *DayTime*, *NightTime* or *Weekend*.
- **Locator**: string value containing the locator of the communication channel such as the telephone number or email address. Where possible URL syntax should be used for the value of this key.

Annex A

(informative)

Common Metadata Hierarchy

A.1 Registered Second Class Name Prefixes

Prefix	Organization
<i>GTS_</i>	NPES & ISO
<i>CIP4</i>	CIP4

Annex B

(informative)

Metadata Encoding Examples

B.1 XML metadata encoding

Examples using the PPML METADATA element:

```
<METADATA Creator="ACME">
  <DATUM Key="CIP4:Root">
    <Metadata xmlns="urn:cip4.org:CommonMetadata:CIP4"
xmlns:ACME="urn:cip4.org:CommonMetadata:ACME">
      <Conformance>base</Conformance>
      <Creator>ACME</Creator>
      <ModificationDate>20090723T111423+01:00</ModificationDate>
      <ACME:ProductVersion>1.1.2</ACME:ProductVersion>
      <ACME:Status>Softproof</ACME:Status>
    </Metadata>
  </DATUM>
</METADATA>
```

```
<METADATA Creator="ACME">
  <DATUM Key="CIP4:Root">
    <Recipient xmlns="urn:cip4.org:CommonMetadata:CIP4"
xmlns:ACME="urn:cip4.org:CommonMetadata:ACME">
      <UniquelId>123456<cip4:UniquelId>
      <Contact>
        <Person>
          <FirstName>Stefan</FirstName>
          <LastName>Daun</LastName>
          <Organisation>Fraunhofer IGD</Organisation>
          <Department>A3</Department>
        </Person>
        <Address>
          <AddressLines>
            <Item>Fraunhoferstr. 5</Item>
            <Item>64283 Darmstadt</Item>
            <Item>Germany</Item>
          </AddressLines>
          <StreetName>Fraunhoferstr.</StreetName>
          <CivicNumber>5</CivicNumber>
          <PostalCode>64283</PostalCode>
          <City>Darmstadt</City>
          <Country>Germany</Country>
        </Address>
      </Contact>
      <ACME:Gender>Male</ACME:Gender>
      <ACME:Offer>Discount10</ACME:Gender>
    </Recipient>
  </DATUM>
</METADATA>
```

CIP4

Examples using the PPML PRIVATE_INFO element (PPML used in ANSI CGATS.20:2002 and ISO 16612-1):

```
<PRIVATE_INFO Creator="ACME">
  <Root xmlns="urn:cip4.org:CommonMetadata:CIP4"
  xmlns:ACME="urn:cip4.org:CommonMetadata:ACME">
    <Metadata>
      <Conformance>base</Conformance>
      <Creator>ACME</Creator>
      <ModificationDate>20090723T111423+01:00</ModificationDate>
      <ACME:ProductVersion>1.1.2</ACME:ProductVersion>
      <ACME:Status>Softproof</ACME:Status>
    </Metadata>
  </CIP4_Root>
</PRIVATE_INFO>
```

```
<PRIVATE_INFO Creator="ACME">
  <Root xmlns="urn:cip4.org:CommonMetadata:CIP4"
  xmlns:ACME="urn:cip4.org:CommonMetadata:ACME">
    <Recipient>
      <Uniqueld>123456<cip4:Uniqueld>
      <Contact>
        <Person>
          <FirstName>Stefan</FirstName>
          <LastName>Daun</LastName>
          <Organisation>Fraunhofer IGD</Organisation>
          <Department>A3</Department>
        </Person>
        <Address>
          <AddressLines>
            <Item>Fraunhoferstr. 5</Item>
            <Item>64283 Darmstadt</Item>
            <Item>Germany</Item>
          </AddressLines>
          <StreetName>Fraunhoferstr.</StreetName>
          <CivicNumber>5</CivicNumber>
          <PostalCode>64283</PostalCode>
          <City>Darmstadt</City>
          <Country>Germany</Country>
        </Address>
      </Contact>
      <ACME:Gender>Male</ACME:Gender>
      <ACME:Offer>Discount10</ACME:Offer>
    </Recipient>
  </Root>
</PRIVATE_INFO>
```

B.2 PDF metadata encoding

Examples using PS/PDF encoding using DPM in ISO PDF/VT (please refer to Annex C of the ISO PDF/VT standard for additional examples):

```
/DPM <<
  /CIP4_Root <<
    /CIP4_Metadata <<
      /CIP4_Conformance (base)
      /CIP4_Creator (ACME)
      /CIP4_ModificationDate (20090723T111423+01:00)
      /ACME_ProductVersion (1.1.2)
      /ACME_Status (Softproof)
    >>
  >>
>>
```

```
/DPM <<
  /CIP4_Root <<
    /CIP4_Recipient <<
      /CIP4_Uniqueid (123456)
      /CIP4_Contact <<
        /CIP4_Person <<
          /CIP4_FirstName (Stefan)
          /CIP4_LastName (Daun)
          /CIP4_Organisation (Fraunhofer IGD)
          /CIP4_Department (A3)
        >>
        /CIP4_Address <<
          /CIP4_AddressLines [
            (Fraunhoferstr. 5)
            (64284 Darmstadt)
            (Germany)
          ]
          /CIP4_StreetName (Fraunhoferstr.)
          /CIP4_CivicNumber (5)
          /CIP4_PostalCode (64283)
          /CIP4_City (Darmstadt)
          /CIP4_Country (Germany)
        >>
      >>
      /ACME_Gender /Male
      /ACME_Offer (Discount10)
    >>
  >>
>>
```