TECHNICAL REPORT

# ISO/IEC TR 10183-1

First edition
1993-04-01

# Information technology — Text and office systems — Office Document Architecture (ODA) and interchange format — Technical Report on ISO 8613 implementation testing —

## Part 1:
Testing methodology

*Technologies de l'information — Bureautique — Architecture de documents de bureau (ODA) et format d'échange — Rapport technique sur l'essai de mise en œuvre de l'ISO 8613 —*

*Partie 1: Méthodologie d'essai*

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

Reference number
ISO/IEC TR 10183-1:1993(E)

## 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.

The main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 10183-1, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 10183 consists of the following parts, under the general title *Information technology — Text and office systems — Office Document Architecture (ODA) and interchange format — Technical Report on ISO 8613 implementation testing*:

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

© ISO/IEC 1993
All rights reserved. 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

— *Part 1: Testing methodology*
[Technical Report]

— *Part 2: Framework for abstract test cases*

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

This page intentionally left blank

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

# Information technology – Text and office systems – Office Document Architecture (ODA) and interchange format – Technical Report on ISO 8613 implementation testing

## Part 1:
Testing methodology

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

## 1 Scope

This part of ISO/IEC TR 10183 defines a testing methodology and provids a framework for specifying abstract test cases for ISO 8613 implementation testing, the overall objective being the provision of a suitable base for testing the *interworking* capability of ODA implementations.

Such testing will assist in the analysis of an implementation's ability to interwork in an environment of implementations of ISO 8613 and implementations of International Standardized Profiles (ISPs) based on ISO 8613. ISPs are standardized document application profiles that have been internationally harmonized. As such they represent agreed stabilized subsets of ISO 8613 designed for the interworking of ODA systems at different levels of functionality.

In ISO 8613 the term "conformance" refers to the conformance of a data stream to the rules specified in ISO 8613. This includes the conformance of a data stream to a document application profile based on ISO 8613. Conformance testing methodology as defined in ISO 8613-1 Annex G covers the analysis of data streams without regard to the capabilities of implementations to generate or receive conforming data streams. To achieve an environment of interworking ODA systems, it is necessary to have a testing methodology that can verify implementation support for ISO 8613 and DAP's at the semantic level as well as the data stream or syntax level.

Hence, implementation testing is additional testing that supplements the conformance testing of data streams. It increases the probability that different implementations of ISO 8613 and ISPs are able to interwork. Implementation testing is based on the concept of measuring an implementation's ability to generate and/or receive a representative set of documents. If an implementation can exhibit this capability, then it is more likely to successfully interwork with other verified implementations exchanging a wider range of documents.

The implementation testing methodology introduces the requirement for abstract test cases as well as procedures for their use in the generation and reception testing of implementations.

In establishing a framework for implementation testing, a conceptual model of ODA systems has been developed to describe, in an abstract sense, the multitude of configurations and limitations of real ODA systems.

The methodology contained in this technical report caters for the testing of implementations of ISP's based on ISO 8613. The methodology may also be used for testing other Document Application Profiles based on ISO 8613.

This part of ISO/IEC TR 10183

- defines terms used in the context of Implementation testing;
- presents a conceptual model of ODA implementations;
- gives a phased approach to testing implementations;
- determines the functional components in generation and reception testing;
- gives the requirements for abstract test cases.

This part of ISO/IEC TR 10183 does not cover the testing of user interfaces in an ODA based system. Any suitable system interface is only used as a point of control and observation to verify that ODA document transformations associated with the various ODA processes have been carried out as claimed by an implementor.

Where appropriate, concepts and terminology described in ISO 9646 have been used in this Technical Report. In some cases, definitions and concepts have been adapted to cater for the fact that ODA is not an OSI protocol.

## 2 References

ISO 8613-1:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 1: Introduction and general principles.*

ISO 8613-2:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 2: Document structures.*

ISO 8613-4:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 4: Document profile.*

ISO 8613-5:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 5: Office Document Interchange Format (ODIF).*

ISO 8613-6:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 6: Character content architectures.*

ISO 8613-7:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 7: Raster graphics content architectures.*

ISO 8613-8:1989, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 8: Geometric graphics content architectures.*

ISO/IEC 8613-9:–[1], *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 9: Audio content architecture.*

ISO/IEC 8613-10:1991, *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Part 10: Formal specifications.*

ISO/IEC 9646-1:1991, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts.*

ISO/IEC 10183-2:–[1], *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format – Testing methodology and abstract test cases – Implementation testing methodology – Part 2: Abstract test suites.*

ISO/IEC ISP 10610-1:–[1], *Information technology – International Standardized Profile FOD11 – Office Document Format – Simple document structure – Character content architecture only – Part 1: Document application profile.*

ISO/IEC ISP 11181-1:–[1], *Information technology – International Standardized Profile FOD26 – Office Document Format – Enhanced document structure – Character, raster graphics and geometric graphics content architectures – Part 1: Document application profile.*

ISO/IEC ISP 11182-1:–[1], *Information technology – International Standardized Profile FOD36 – Office Document Format – Extended document structure – Character, raster graphics and geometric graphics content architectures – Part 1: Document application profile.*

1) To be published.

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

## 3 Definitions

For the purposes of this part of ISO/IEC TR 10183, the definitions given in ISO 8613-1 and the following definitions apply.

NOTE 1 Some terms from ISO/IEC 9646-1 are also used where there is an equivalence of meaning.

**3.1 ODA document:** An amount of information structured in accordance with the abstract architecture model defined in ISO 8613.

**3.2 ODA data stream:** An ODIF or ODL data stream in which the data elements representing constituents and attribute values are in accordance ISO 8613-1:1991, clause 8, and any referenced standard.

NOTE 2 This definition requires, for example, that all constituent references must be satisfied either in the data stream or by some external reference.

**3.3 encode:** A transformation from a local system representation of an ODA document to an ODA data stream.

**3.4 originate (generate) a document:** Provide an ODA data stream for interchange.

**3.5 ODA interchange process:** An ODA document is originated, transmitted by means of data communications or the exchange of storage media and received.

**3.6 receive a document:** Accept a data stream as a representation of an ODA document.

**3.7 decode:** A transformation from an ODA data stream to a local system representation of an ODA document.

**3.8 DAP-L data stream:** An ODA data stream in which the data elements are in accordance with clause 7 of a particular DAP level "L", defined in accordance with ISO 8613-1.

**3.9 Implementation Under Test (IUT):** An implementation of ISO 8613 being that part of a real open system which is to be studied by testing.

**3.10 ODA Implementation:** An implementation that can generate a representative set of ODA data streams and/or receive and possibly further process a representative set of ODA data streams.

NOTE 3 Support for ODA generation, reception, and further processing depends on implementation characteristics (e.g. converter, editor, printer etc.).

**3.11 ISP Implementation:** An ODA implementation that can generate a representative set of ISP DAP-L data streams and/or receive and possibly further process a representative set of ISP DAP-L data streams.

NOTE 4 Support for ISP DAP-L generation, reception and further processing depends on implementation characteristics (e.g. converter, editor, printer etc.), as defined by a GSS/RSS (3.22).

**3.12 edit:** A transformation from a local system representation of a processable form ODA document to a local system representation of a revised version of that processable form ODA document. This transformation being in accordance with the editing process part of the ODA document processing model.

NOTE 5 *Edit* includes the "create" transformation from a null document to a processable document. *Edit* is a particular case of the *modify* transformation (3.13).

**3.13 modify:** A transformation from a local system representation of an ODA document to a local system representation of a revised version of that ODA document.

NOTE 6 *Modify* is a transformation relating to any changes to the document structures, constituents, attributes and content in a local system representation of an ODA document. An example of *modify* is the transformation that results from deleting the specific layout structure of a formatted processable form document to create a processable form version of that ODA document. *Modify* includes but is not limited to the *edit* transformation (3.12).

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

**3.14 layout:** A transformation from a local system representation of a processable form ODA document to a local system representation of a formatted form ODA document. This transformation being in accordance with the layout process part of the ODA document processing model.

**3.15 image:** A transformation from a local system representation of a formatted or formatted processable form ODA document to a human-perceptible representation of that ODA document. This transformation being in accordance with the imaging process part of the ODA document processing model.

NOTE 7 *Image* is a particular case of the *view* transformation (3.16).

**3.16 view:** A transformation from a local system representation of an ODA document to a human-perceptible representation of that ODA document.

NOTE 8 Support for the *view* transformation in an implementation depends on the presentation medium (e.g. paper, screen, etc.) and the application (e.g. printer, browser, archival system etc.). *View* includes but is not limited to the *image* transformation (3.15).

**3.17 local system representation:** That part of an SUT which represents an ODA document usually in a system dependent format.

**3.18 ODA/ISP based implementation:** An ODA or ODA ISP implementation (3.10, 3.11).

**3.19 ODA/ISP implementation testing:** Testing the extent to which an IUT can support the functional elements of ODA or ODA ISPs.

**3.20 ODA/ISP based system:** The real open system in which an ODA or ODA ISP implementation exists.

**3.21 Generating and Receiving Support Statement Proforma (GSS/RSS Proforma):** Part of an ISP that details the generation, reception and further processing support requirements/options applicable to implementations of that ISP's DAP.

**3.22 Generating and Receiving Support Statement (GSS/RSS):** A statement that details the generation, reception and further processing support claimed by an implementation for a particular Document Application Profile of an ISP.

**3.23 ODA Interworking:** The ability of an implementation to generate and/or receive and possibly further process ODA data streams as specified in a GSS/RSS.

NOTE 9 "Further process" includes one or more of the transformations defined *Edit*, *Modify*, *Layout*, *Image* and *View* transformations defined in 3.12, 3.13, 3.14, 3.15, 3.16.

**3.24 point of control and observation:** A point within a test environment at which control and observation is specified by a test case.

**3.25 system under test (SUT):** The real open system in which the IUT resides.

**3.26 test case purpose:** A natural language description of the objective of an abstract test case, focusing on a single functional element (e.g. verifying the support of a specific value of an ODA attribute).

**3.27 abstract test case (ATC):** A test case purpose together with a specification of any ODA (or ODA ISP) functional elements necessary to achieve the test case purpose and assign a test verdict.

**3.28 executable test case:** A realization of an abstract test case.

**3.29 ODA/ISP test document:** An ODA or ODA ISP conforming document that contains one or more abstract test cases.

**3.30 abstract test suite:** The complete set of test cases needed for testing an implementation's support for ODA or ODA ISPs.

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

## 4 Abbreviations

**DAP**: Document Application Profile.

**ISP**: International Standardized Profile.

**IUT**: Implementation Under Test.

**PCO**: Point of Control and Observation.

**Pc**: Process component.

**Sc**: System interface component.

**Ic**: Interchange component.

**GSS**: Generating Support Statement.

**RSS**: Receiving Support Statement.

**DUT**: Document Under Test

**TCS**: Test Case Specifications (used in generation testing)

**TDS**: Test Data Streams (used in reception testing)

**FDAR**: A local representation of a formatted form (FDA) ODA document

**PDAR**: A local representation of a processable form (PDA) ODA document

**FPDAR**: A local representation of a formatted processable form (FPDA) ODA document

ECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

## 5 General concepts of implementation testing

The purpose of the ODA standard, *"facilitating the interchange of documents"*, will not be completely achieved until implementations can be tested to determine whether they can interwork correctly. Standard test suites are required for use by suppliers or implementors in self-testing, by users of ODA products, by third party testing organizations and other administrations concerned with ODA implementation testing.

ODA implementations can take many forms from gateways that receive and regenerate data streams with little processing, to full ODA document processing systems that implement all the processes facilitated by ISO 8613. ODA implementation testing involves testing a system's capability to support features identified in the DAP part of the ODA ISPs. This is carried out by using the relevant ISP Abstract Test Suite (ATS) and checking what is observed against both the support requirements of the ISP (GSS/RSS Proforma) and the implementor's statement regarding the implementation's capabilities (GSS/RSS). Not all ODA implementations will be ISP implementations. However, the methodology, framework and procedures specified in this Technical Report can be applied to any Document Application Profile (DAP) defined in accordance with the Document Application Profile Proforma and Notation defined in ISO 8613-1. A "full ODA implementation", that is, an ODA implementation that supports the generation and reception of any ODA data stream can be considered as an unconstrained DAP.

Along with this testing methodology and the framework for abstract test cases,specified in ISO/IEC TR 10183-2, it is necessary to define procedures that have to be followed during ODA implementation testing. This should lead to comparability and wide acceptance of test results produced by different testers, and thereby minimize the need for repeated implementation testing of the same system.

The purpose of ODA implementation testing is twofold. Firstly, it increases the probability that an implementation can interwork and secondly it gives users an increased level of confidence in tested ODA products even though implementation testing can only provide an indirect assessment of a product's overall capability.

ODA and ODA ISP's are complex standards and exhaustive testing of all combinations of features is impractical on both technical and economic grounds. However, implementation testing does show that an implementation can demonstrate the required capabilities for a representative set of test cases.

Since the functionality of ODA implementations can be separated into generation and reception of data streams, the methodology, framework for abstract test cases and procedures for testing are similarly divided to cater for the two scenarios. For generation testing, the overall methodology consists of an implementation being provided with a set of test cases representing aspects of the functionality level corresponding to a particular ISP. The implementation is then required to generate a number of data streams that are tested by a tester for both data stream conformance to an ISP and for presence of the required functionality. For reception testing, a set of documents containing test cases that represent the functionality level corresponding to a particular ISP are passed to the implementation. The implementation is then required to process the data streams in accordance with the support requirements of the ISP and the implementor's statement regarding the support capability of the implementation. The tester then analyses at points of observation, document output from the implementation. In both generation and reception testing, a test report, is produced which records the test verdicts. In both situations, an analysis of the test reports is carried out to determine if the outcomes have matched the implementor's statement regarding the implementation's capabilities. In this way, the implementation testing is intended to give a clear statement about an implementation's capability of interworking based on experience with a set of test cases.

## 6 Conceptual model of ODA based systems

The definition of a conceptual model of the systems under test (SUTs) is an important element in establishing a testing framework for implementation testing since the development of a testing methodology and test methods are obviously related to the various configurations and limitations of real systems. Since ODA documents are information structures, they can be used in numerous application contexts. A conceptual model for ODA based systems is useful for categorizing such systems.

NOTE 10 An ISP based system is considered as a particular instance of an ODA based system.

An ODA based system can be seen as consisting of three conceptual components for a typical system that originates and/or receives ODA documents:

a) the interchange component (Ic), represents the functionality of transferring ODA data streams in and out of the system;

b) the process component (Pc), represents that part of an application that performs transformations on the local system representation of an ODA document;

c) the system interface component (Sc), represents the mechanisms used to control and observe transformations on the local representation of an ODA document.

The components above are conceptual and do not necessarily reflect real system implementation structure. Component boundaries are virtual boundaries that exist only in the representation of a real ODA based system.

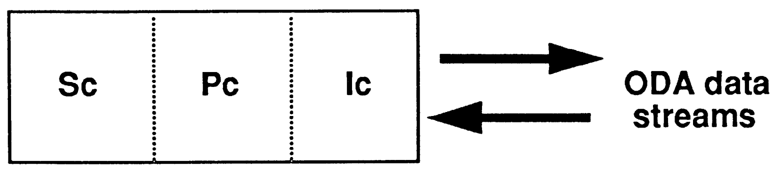



Figure 1 - ODA based system conceptual model

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

### 6.1 The interchange component

The interchange format of an ODA data stream is not generally suitable for application processing. Systems wanting to process an ODA document will normally convert to and from the interchange format to some form of local representation that allows easier access to and manipulation of the document structures, constituents, attributes and content. The interchange component of an ODA based system deals with this aspect.

Three representation levels can be visualized:

a) The system input/output level, originates and receives the ODA data streams either as operating system files, MHS message body parts, files on magnetic media, files on network file systems or received through file transfer protocols etc. At this level the ODA data streams remain uninterpreted.

b) The next level, the encode/decode level, deals with the transfer syntax of the ODA data streams. An ODA data stream is created/interpreted according to the prescribed ASN.1 grammar and ISO 8613-5 rules, thus building or extracting the ODA constituents one by one.

c) The last level, the so-called externalize/internalize level, is often performed in conjunction with the previous level. At this level, the created/interpreted constituents are converted to and from internal data structures appropriate for the application(s) of the local ODA based system.

In a real system, the last two levels are often integrated with the application(s) of the process component but may be distinct parts of the process component together with the application, as in the case of an ODA toolkit.

### 6.2 The process component

This is the application within the local system that can support ODA semantics, working from the elements making up the local representation of the ODA document.

The aspects of the application of interest in the context of this model are those which carry out the *modify*, *layout* and *view* transformations on ODA documents. Implementations may wish to claim support for these types of transformations and they will subsequently need to be tested. The test cases for such applications will be developed to cater for ODA systems that are originators and/or receivers and within these sets of applications, for those that can carry out one or more of the identified transformations.

A process component may support one or more of the ODA semantics, which can be viewed as a set of features (or properties) supported; each feature consisting of a set of ODA elements (attributes, constituents) together with a description of an abstract effect to be achieved by the processing component using these elements (not how this is carried out). Such a feature can be as simple as being able to accomplish character text emphasis or as complex as producing a specific layout structure or displayed document from a processable form ODA document. The set of ODA elements can therefore be a single attribute value and semantic, but will often consist of more than one ODA element and associated semantic. An ODA data stream is viewed therefore as carrying embedded features put there by one document process and requiring interpretation by another document process.

### 6.3 The system interface component

The applications used in an ODA based system are not necessarily end-user applications, i.e. applications that interact with users such as editors, document filing systems etc. They will also include applications such as document print servers or automatic translators which convert documents to and from ODA.

This component represents the various mechanisms available for controlling transformations within an ODA based system and the mechanisms used in the ODA based system to observe results. For example, where an application translates from a non-ODA document to an ODA document, the controlling mechanism could be that non-ODA system's user interface. Where the ODA based system is a print server, the observed result might occur through an imaged paper document.

IECNORM.COM: Click to view the full PDF of ISO/IEC TR 10183-1:1993

# 7 Implementation Testing Methodology

To determine if ODA/ISP implementations can interwork correctly, it is necessary to control and observe the implementation when it is subjected to testing. The testing methodology is based on the approach of requesting an implementation to demonstrate its capability to support specific ODA or ISP features by using, in generation or reception, a set of test cases containing specific ODA/ISP features. ODA data streams received from a generating system or output from a receiving system can then be analyzed to determine if the claimed support can be verified. Since ODA/ISP applications are diverse the way that control and observation is performed may vary even when different implementations claim to support the same feature set.

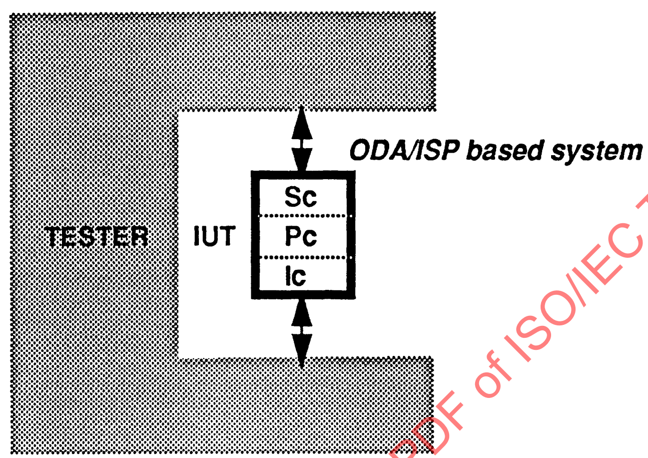



Figure 2 - ODA/ISP based system testing architecture

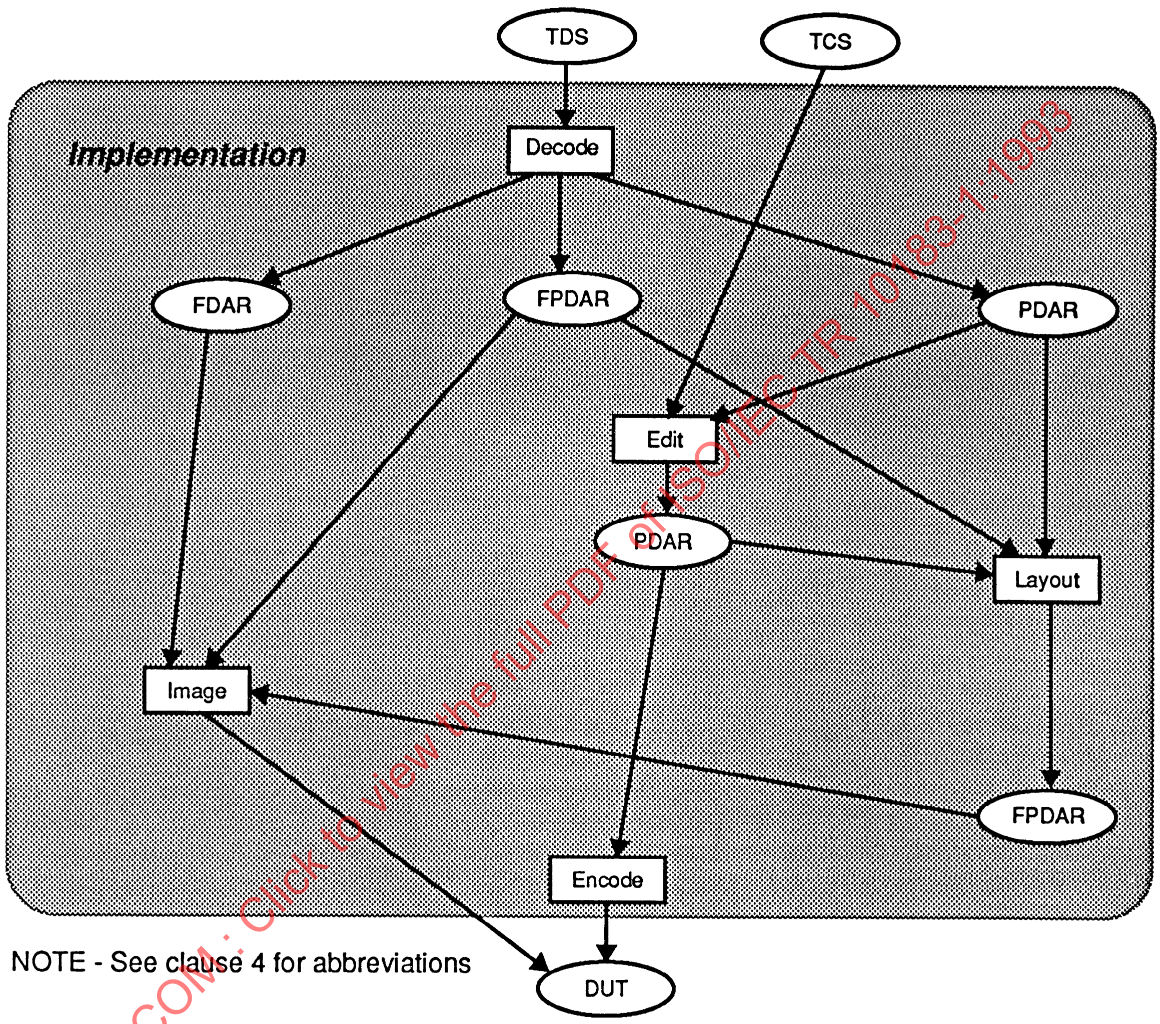
IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

As described in clause 5, IUTs are provided with a set of test cases to test the generation and reception capabilities of the implementation. These take the form of Test Case Specifications (TCS) for generation testing and Test Data Streams (TDS) representing ODA/ISP documents, containing test cases, in the case of reception testing. The IUT provides documents for analysis either in the form of data streams (for generation testing) and/or documents observable by a suitable system interface (for reception testing).

## 7.1 Processes and data structures in an ODA/ISP implementation

In order to have a better understanding of the scope of the ODA/ISP implementation testing methodology it is useful to consider some of the different implementation processes and data structures involved (see figure 3). Having received a test data stream (TDS) or a set of test case specifications (TCS), an implementation under test must provide, for the tester, a set of data streams in the case of generation testing and a set of documents, possibly in various representations, in the case of reception testing. Both these forms are termed "Document Under Test" (DUT). The class of implementations that receive and generate the TDS,TCS and DUT can be categorized in terms of the different paths of processes and data structures used by that system. One example path might be for a class of implementations that accept a "Test Data Stream" representing a formatted form ODA document. Such a class of implementations will decode that "Test Data Stream" into a local representation (FDAR). Subsequently, some of these implementations might take the local representation and pass it through an local imaging process to create a human-perceptible form of the ODA document. Such a document could be used to observe support, through the *image* transformation, for specific presentation features. It should be noted that the imaged document is only one of the possible representations that a DUT may take.

There are a number of possible implementations and figure 3 shows only those processes and data structures associated with a particular class of ODA implementation. Each data structure is represented by an oval and each process by a box. Further processes and data structures, not shown in figure 3, may be used by an implementation or be useful as additional elements of the system interface component. One example is that an implementation may support the *modify* transformation from a formatted processable form document to a processable form document. Another example, might be an implementation that provided a "browser" application that supported a *view* transformation to give a human-perceptible form of the structure or content of a processable form document representation (PDAR).




NOTE - See clause 4 for abbreviations

Figure 3 - Example set of implementation processes and data structures

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

The example in figure 3, shows a class of implementations that only support the generation of processable form documents although formatted, processable and formatted processable form documents can be received and further processed. From this class of implementation shown, a number of possible paths can be identified:

1 TCS-Edit-PDAR-Encode-DUT
2 TDS-Decode-FDAR-Image-DUT
3 TDS-Decode-FPDAR-Image-DUT
4 TDS-Decode-PDAR-Edit-PDAR-Encode-DUT

5 TDS-Decode-PDAR-Layout-FPDAR-Image-DUT
6 TDS-Decode-FPDAR-Layout-FPDAR-Image-DUT
7 TDS-Decode-PDAR-Edit-PDAR-Layout-FPDAR-Image-DUT

These are candidate test paths, for this class of implementation, each requiring an associated set of test cases.

### 7.2 The initial scope of implementation testing

The initial scope of any ODA implementation testing needs to concentrate on the most likely types of implementation in order to reduce the effort required to develop the necessary sets of test cases. Developing test cases for all possible types of implementation and for the full range of ISO 8613 functionality is clearly an impractical if not impossible task. By scoping test case development, test cases can be developed in a timely fashion that will give a sufficient degree of confidence that a verified implementation can interwork successfully when presented with a broader range of documents. A further restriction in scope needs to be based on the rationale of there being more ISP implementations of a particular type. For example, translators to and from proprietary document processing systems are expected to be more prevalent, in the short term, than document processing systems supporting the full ISP architecture that would, for example, facilitate class-driven editing applications.

It follows that the development of test cases for the first versions of ISP should therefore focus on those features more likely to be supported by expected products. That is, ISP implementations translating to and from proprietary document processing systems. For example, test cases to verify an implementation's support for the document features specifically detailed in clause 6 of the ISP DAPs.

## 8 Generation testing

Figure 4 shows the functional components necessary for the testing of data streams generated by an IUT for conformance to an ISP at a particular functional level DAP-L. Figure 4 shows three domains "Standard", "Implementation" and "Tester". These domains delineate the responsibilities for the production of information/data structures etc. used in ISP generation testing.

The Standard domain shows which information is necessary from the standards world to enable ISP implementation generation testing. The implementation domain shows that an implementation is responsible for completion of a Generating Support Statement (GSS), the provision of additional implementation specific information required for testing and the production of a number of ODA data streams to be tested when provided with the abstract test cases for an ISP DAP-L. The tester domain shows which information will be used by a test system when analyzing the ODA data streams produced by the

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

implementation.

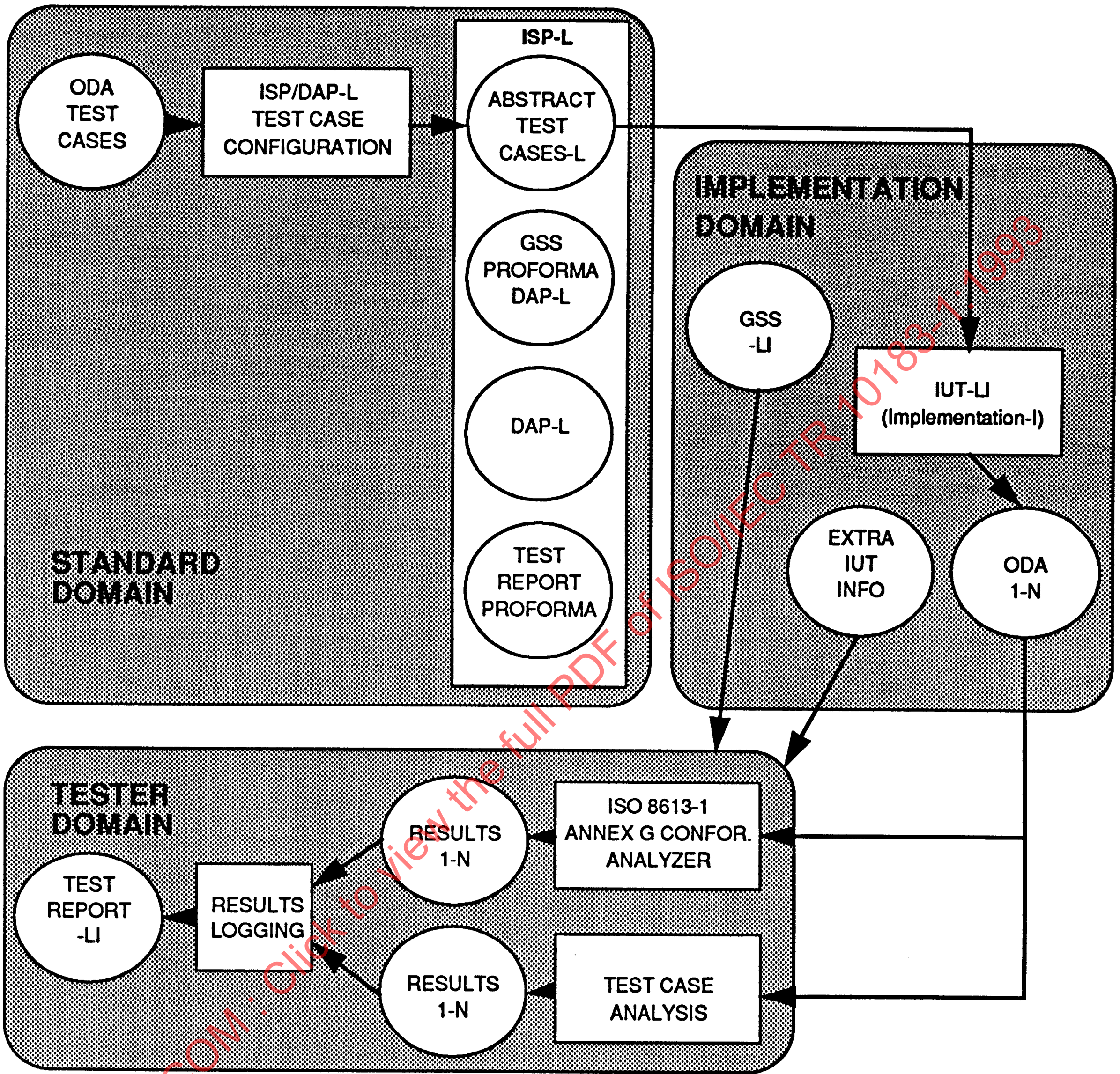



Figure 4 - Functional Components in Generation Testing

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993

### 8.1 Standard Domain

In the Standard Domain, ODA generation test cases are required, from which generation test cases for a particular DAP at functional level L can be configured. In order that a standard approach is given to the implementor generating support statement, a proforma is standardized for the ISPs. This is termed the GSS proforma for DAP-L. This proforma is being developed for part 2 of the ISPs by the harmonization activity that is proposing ISPs to ISO. Part 1 of the ISP is the DAP at a particular functional level L. There will also be a need to develop a test report proforma for the tester domain that provides a common format for generation testing results.

### 8.2 Implementation Domain

The implementation domain for generation testing takes the abstract test cases for a particular DAP at functional level L and creates from a given implementation a set of test data streams (ODA 1-N) for analysis by the tester. The tester domain also needs a completed GSS for the DAP-L as well as any extra implementation information necessary for use in generation testing.

### 8.3 Tester Domain

The tester domain shows that a tester uses the rules in ISO 8613-1 Annex G Conformance Testing Methodology to analyze the ODA data streams 1-N for conformance to ISO 8613 and the relevant DAP. The set of ODA data streams are also used to test for the presence of the test cases in accordance with the GSS-LI information. In both cases the results 1-N for each data stream are passed to a results logging process that provides the verdicts in a test report for that particular implementation. Note that there is not necessarily a one-to-one correspondence between test cases and data streams and that not necessarily every test case needs to be present in the set of data streams provided by a given implementation I.

## 9 Reception testing

Figure 5 shows the functional components necessary for the testing of documents produced by an IUT for conformance to an ISP at a particular functional level L. Figure 5 shows three domains “Standard”, “Implementation” and “Tester”. These domains delineate the responsibilities for the production of the information/data structures etc. used in ISP reception testing.

The standards domain shows which information from the standards world is necessary to enable ISP implementation reception testing. The implementation domain shows that it is responsible for completion of a Receiving Support Statement (RSS), the provision of additional implementation specific information required for testing and the production of a number of documents (possibly in one or more representations) to be tested when provided with a set of data streams for ISP functional level L implementations. The tester domain shows which information will be used

IECNORM.COM : Click to view the full PDF of ISO/IEC TR 10183-1:1993