

AD-A261 193

①

NBSIR 88-3726



A Collection of Technical Studies Completed for the Computer-aided Acquisition and Logistic Support (CALs) Program Fiscal Year 1987 Volume 1 of 4

Sharon J. Kemmerer, Editor

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Sciences and Technology
Information Systems Engineering Division
Gaithersburg, MD 20899

March 1988

SDTIC
ELECTE
FEB 25 1993
E D



Remembering America's Progress
1913-1988

U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

93-03916



4258

Approved for public release
Distribution Unlimited

98 2 24 040

NBSIR 88-3726

**A COLLECTION OF TECHNICAL STUDIES
COMPLETED FOR THE COMPUTER-AIDED
ACQUISITION AND LOGISTIC SUPPORT
(CALS) PROGRAM
FISCAL YEAR 1987
VOLUME 1 OF 4**

Sharon J. Kemmerer, Editor

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Sciences and Technology
Information Systems Engineering Division
Gaithersburg, MD 20899

March 1988

Account for	
NTIS	X
DTIC	
U.S. GPO	
Justification	
By	
Date	
Approved for Release	
Dist	Availability
A-1	

U.S. GOVERNMENT PRINTING OFFICE: 1987

U.S. DEPARTMENT OF COMMERCE, C. William Verity, Secretary
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

EXECUTIVE SUMMARY

The overall objective of the Department of Defense Computer-aided Acquisition & Logistic Support (CALs) Program is to integrate the design, manufacturing, and logistic functions through the efficient application of computer technology. CALs is a program to apply existing and emerging communications and computer-aided technologies in DoD and industry to:

- o Integrate and improve design, manufacturing, and logistic functions; thereby bridging existing "islands of automation."
- o Actively influence the design process to produce weapon systems that are more reliable and easier to support and maintain.
- o Shift from current paper-intensive weapon support processes to a highly automated mode of operation, based on a unified DoD interface with industry for exchange of logistic technical information in digital form.

The CALs program was established by the Deputy Secretary of Defense in September 1985 to implement the recommendations of a Joint Industry/DoD Task Force. Management is provided by a DoD Steering Group, an OSD CALs Policy Office, and their counterparts in each Military Department and the Defense Logistics Agency. The CALs Policy Office has obtained the support of the National Bureau of Standards in the selection and implementation of CALs standards. An Industry Steering Group has also been established to focus the work of key industrial associations and the defense contractor community in CALs implementation.

The Bureau has been funded since Spring 1986 to recommend a suite of industry standards for system integration and digital data transfer, and to accelerate their implementation. NBS activities during 1986 were primarily aimed at:

- o familiarizing NBS technical staff with key DoD logistic functions and CALs demonstration projects,
- o briefing DoD personnel, contractors, and other interested parties on Federal, national, and international standardization efforts that are expected to support CALs objectives,
- o identifying a preliminary set of standards required for data interchange in support of CALs, and
- o developing reports on the four broad categories of standards required to support the interchange of CALs digitized technical information: (1) product definition data, (2) graphics, (3) text, and (4) data management.

As a result of these efforts, NBS made a preliminary identification of several high-priority standards implementations

needed for CALS data interchange and access.¹ Building on knowledge and experience gained during FY86, NBS focused on the following activities in FY87: developing a CALS Framework, Development Plan and Core Requirements Package; providing technical support for standards development and implementation; and conducting workshops and meetings to promote dialogue with the Services, the Defense Logistics Agency, and industry.

A major FY87 thrust was the completion of initial documentation of the high-priority standards required in the CALS environment. Some of these standards (e.g., SGML, IGES) required tailoring or enhancement. Other standards required a "push" (e.g., CGEM) for their development in a timely fashion. These four volumes are a collection of the final reports presented to the CALS Policy Office.² The collection is divided as follows:

VOLUME 1:

Text

- Evaluation of Text Interchange Methods
- Plan for Conformance Testing for DoD Implementation of SGML
- Guidelines for the Development of Tags for SGML
- The NBS FIPS - SGML Validation Suite
- The NBS FIPS - SGML Reference Parser
- Using SGML - Application Guidelines
- ODA/ODIF Implementation Agreement a Document Application Profile

Data Management

- CALS Report on Data Management Standards
- Supporting Logistic Support Analysis (LSA) Using the Information Resource Dictionary System (IRDS)

Media

- ICST Recommendations on Optical Disks and Interface Requirements for Planned EDMICS Procurement, Final Report

¹ Kemmerer, S., Editor, "Final NBS Report for CALS, FY86," U.S. Department of Commerce, National Bureau of Standards, NBSIR 87-3566, May 1987.

² The publishing of this collection of reports does not imply the CALS Policy Office has endorsed the conclusions and recommendations presented.

Raster Compression
Report on Raster Graphics

Tiled Raster Interchange Format, TRIF Version 1.0, Rev. 1.2

Conformance Testing

NBS Plan for Validation (Conformance Testing) of Computer
Products in Support of the CALS Program

VOLUME 2:

Graphics

Raster-to-Vector Conversion: A State-of-the-Art Assessment

Development of CGM Validation Routines

CALS Application Profile for CGM

CALS Requirements Reflected in the Extended CGM (CGEM)
Standards Effort

A Reference Implementation for CGM, Functional Requirements
and Conceptual Design

IGES to CGM Translator Design Specification

VOLUME 3:

Graphics

CGM Registration For CALS Requirements

VOLUME 4:

Product Data

Guidelines for Testing IGES Translators

Guidelines for IGES Application Subsets

The following are additional deliverables completed by NBS during FY87 but under separate cover. They are available through the CALS Policy Office.

CALS Core Requirements, Phase I.0

CALS Framework

CALS Program Integration of Logistic Support Analysis and Reliability and Maintainability Data Deliverables

CALS Current State of Digital Technology (Phase I.0)

CALS Workshop Proceedings:

Graphics Data Interface for Engineering Design and Technical Publication Systems (January 13/14)

Introduction to the Core Requirements Package (April 23)

MILSTD-1840A, Automated Interchange of Technical Information

MILSPEC-D-28000, Digital Representation for Communication of Product Data: Application Subsets

MILSPEC-M-28001, Manuals, Technical: Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange

CONTRIBUTIONS

NBS would like to acknowledge the major technical contributors to this volume. In alphabetical order they are:

Cita Furlani

Alan Goldfine

James Heath

David Jefferson

Arnold Johnson

Sharon Kemmerer

Joseph Maggelot

Mary Mitchell

Frances Nielsen

Mark Skall

Frankie Spielman

Joan Sullivan

TEXT

EVALUATION OF TEXT INTERCHANGE METHODS

PLAN FOR CONFORMANCE TESTING FOR DOD IMPLEMENTATION OF SGML

GUIDELINES FOR THE DEVELOPMENT OF TAGS FOR SGML

THE NBS FIPS - SGML VALIDATION SUITE

THE NBS FIPS - SGML REFERENCE PARSER

USING SGML - APPLICATION GUIDELINES

ODA/ODIF IMPLEMENTATION AGREEMENT
A DOCUMENT APPLICATION PROFILE

15 July 1987

Evaluation of Text Interchange Methods

I. PURPOSE.

To evaluate methods to provide text interchange among differing computer systems. (Task 2.1.3)

II. BACKGROUND.

In the Taft OASD memorandum dated 24 September, 1985, the CALS project was formally initiated, and the decision to use SGML as a minimum was stated. This decision was made primarily because implementations of SGML existed or were in process at the initiation of the project. There are, however, other text interchange standards which should be examined.

During the last decade, there has been a proliferation of computer equipment and software for creating, editing, revising, processing, and transmitting textual information. Unfortunately, users have discovered through painful experience that it is often impossible to exchange information among various makes of equipment. As users become more aware of these problems, they have also become insistent on compatibility in order to avoid being trapped in single-solution dead ends and to avoid paying the price of either converting documents from one product's format to another or rekeying documents.

III. DISCUSSION.

Government agencies, including the DoD, are among the list of users who have paid the price for incompatible text systems and vendors' proprietary software. Now there are a number of standards which can be used for text interchange. Two of these standards, SGML and ODA, are of primary interest to CALS. There is also a defacto text interchange standard, IBM's Document Content Architecture (DCA); however, the major limitation of DCA is its proprietary nature.

What are the specific user, in this case CALS, needs regarding text interchange? The following sections give the requirements for text interchange and address the standards developed to meet those requirements.

Requirements for Text Interchange

CALS applications have a number of general requirements for text interchange, including the need:

- for a standardized way to exchange textual information (documents or parts of documents);
- for the exchanged document to contain a variety of content types, including characters, pictures,

- drawings, images, figures;
- to output the interchanged text on a variety of media (e.g., paper, CRT, laser printer, photocomposer);
 - to pull together parts of documents prepared or processed separately, and conversely, the need to distribute parts of documents for separate processing;
 - for a standardized way to represent the appearance of the document (e.g., multi-column text, various fonts); and
 - to use parts of the interchanged text in database applications.

Standards for Text Interchange

As mentioned above there are three major standards which could be used for text interchange. Computer manufacturers and software suppliers have tended to define their own interchange format standards or have adhered to the defacto standard. The problem with this solution, though, is that it "locks-in" users.

Admitting then a requirement to use a nationally or internationally standardized approach to text interchange, the candidates -- SGML and the Office Document Architecture (ODA) and Interchange Format (ISO 8613) -- can be evaluated. While there are similarities between the functionality of these standards, there are also advantages and disadvantages to using them.

SGML

The SGML standard is a set of rules for defining applications such as a) the structure of document types (e.g., technical reports, technical manuals, training manuals, journals); b) the logical components of the document types independent of the format of those components (e.g., title, section headings, paragraphs); c) references to document contents that cannot be keyed from a keyboard or are external to the document (e.g., special characters, graphics, drawings); and d) specifications for database publishing systems. That is, SGML is a representation language for character text and it can be used for publishing in its broadest definition -- from single medium conventional publishing to multimedia database publishing. The user determines what components to identify within a document and describes those components in SGML.

The basis of SGML is the principle of generic markup of a document; that is, elements of the document are identified to indicate their role (e.g., title) rather than their presentation (e.g., centered, bold). The SGML standard deals primarily with character text and handles it straightforwardly. However, documents, especially technical documents, may contain more than

character text such as scanned images and computer graphics. These are referred to as non-SGML data and SGML merely provides an indication of where this data can be found. This means both the document creator and subsequent user must have access to the non-SGML data and must have agreed a priori to the format of the data.

ODA

On the other hand, ODA defines an explicit document architecture, including a capability for integrating various content types in one document. This document architecture is the form of the information transmitted through a network. ODA relates only to the structure and format of a document in interchange and does not attempt to standardize any processes performed on the document either before or after interchange. This means that the entry, editing, formatting, and internal storage of the document may be different in each system, but before interchange the document is translated into a standardized form which is ODA. Upon receipt, the recipient will translate the document into his own internal format and then process the document. Developers of ODA recognized the inevitable convergence of computer systems and office systems, and therefore decided to develop ODA.

ODA is a method of describing the electronic representation of a document including the types of information found in, or expected to be required in, documents. This structured description is called "architecture." The representation of the document is in a form suitable for interchange between office automation equipment, such as word processors, computer workstations, personal computers, and so on. The encoding of ODA for interchange between such devices is defined in a serial form (called the datastream) that is suitable for use over both current and newly emerging computer communication networks. In particular, ODA uses a standard syntax -- ISO 8824 - Abstract Syntax Notation One (ASN.1) -- defined for use in the Open Systems Interconnection network environment.

Currently three types of content are standardized within ODA. These are character content (any internationally standardized character set can be used), raster graphics content (based on CCITT Group 4 Facsimile), and computer graphics content (based on the CGM standard). Thus, an ODA document could contain facsimile and computer graphics data as well as character text integrated into one consistent datastream.

Status of SGML and ODA

In December 1986, SGML became an International Standard (IS) and is currently proposed as an American National Standard. As a result, more and more implementations of SGML applications have begun to and will continue to appear on the market-place. However, an SGML implementation alone -- without support processes -- is not enough to handle the document. Other

processes to support document creation and to further process (such as format, image) the document need to be defined. Fruition of projects such as the SGML Text-Sensitive Editor, Document Style Specification Language, and the Standard Page Description Language will be integral parts of an SGML text system -- one that handles a document from beginning to end, from idea to print.

In May 1987, the standards community voted to accept a revision of ODA and to rebalot it as a Draft International Standard (DIS) since there were numerous changes to the original DIS. It is expected that ODA will become an IS in February 1988; this February version will be completely aligned with a parallel project in the Consultative Committee for Telegraphy and Telephony (CCITT). This ensures that there will be implementations of ODA on the market soon afterwards. ODA can meet the document interchange need; however, fruition of the Standard Page Description Language will allow a standardized way to image (print or display) ODA documents as well as SGML documents. While needed for the SGML environment, the Document Style Specification Language is not needed in the ODA environment since that functionality already exists as a function of ODA.

In the standards community, efforts have been taken to ensure harmonization of SGML and ODA. Indeed there are applications for the use of both the SGML and the ODA standards, singularly and in combination. It is probable that future text interchange products will focus on accommodating the two. For example, the Digital Equipment Corporation, while recently announcing their new Digital Document Interchange Format (DDIF) based on ODA, also stated the need to handle SGML documents.

Evaluation

The flexibility of SGML to describe any structure discourages rather than encourages interoperability since specific applications of SGML must be defined. Another negative of SGML is the lack of standardization for non-character data. For example, in the CALS environment, compound documents exist. A compound document is a document containing integrated character text, graphics, image data, and possibly other types of information, represented in one datastream. An implementation of SGML could refer to the non-character modules (non-SGML data), e.g., computer graphics content, by the use of entity references; but there would be no guarantee that implementations could successfully exchange this information, since the format would be outside the scope of the SGML representation of the document.

Unlike SGML, ODA integrates various content types, such as character text, facsimile, and computer graphics, in one data stream. (ODA has also been designed to accommodate content types such as audio and video in the future.) In fact, ODA is an architecture for compound document exchange; however, the ODA

standard is less flexible than SGML in that ODA cannot describe any arbitrary structure, but rather must build a structure according to the rules of ODA and according to the standardized content types (character text, raster graphics, and computer graphics).

Both SGML and ODA support revisability of documents, but again the issue of compound documents is raised.

SGML has been used as a tool for database design; however, both SGML and ODA view documents as logical structures of information which allows tools (e.g., databases) to be built to process these structures.

While ODA was developed with interchange in mind, both SGML and ODA can be interchanged over Open Systems Interconnection (OSI) networks. The SGML Document Interchange Format (SDIF) concatenates (packages) SGML files into one datastream for transmission between SGML systems over a network. The Office Document Interchange Format (ODIF) is the form for transmitting ODA documents over any communications networks including OSI networks.

SGML software is available now; only prototypes of ODA software currently exist. However, both SGML and ODA have been, and will continue to be, supported technically and politically at NBS.

The differences between SGML and ODA generally are in 1) application area (publishing versus office), 2) document architecture support (implicit versus explicit), 3) content architecture support, and 4) interchange format encoding. As for point one, it is common knowledge that office and publishing are converging. Today's office equipment -- personal computers, word processors, laser printers -- is capable of very sophisticated publishing-quality techniques. Desktop publishing can be embodied in today's personal computer.

The remaining points have already been discussed. Further information is included in the attached paper from German experts on text interchange.

IV. RECOMMENDATION.

There is a need for conversion and/or translation between SGML encoded documents and the ODA environment. CALS should support this conversion effort. Also, projects such as the Document Style Specification Language and the Standard Page Description Language should be accelerated to accommodate both ODA and SGML environments. These projects should be supported by CALS.

More than the interchange process needs to be addressed by CALS. As mentioned above, SGML and ODA are only part of the solution to interoperability and portability of documents. In particular,

the Standard Page Description Language would ensure that an SGML or an ODA document is imaged as intended by the document's creator. And for SGML systems, the Document Style Specification Language and processes to link the style specification with the marked up document need to be in place.

V. CONCLUSION.

Overall, standards for document interchange are to provide interoperability between different components and then to ensure competitive costs among the vendors of those components. In addition, a direct side effect of using standards versus proprietary solutions is that the standards facilitate procurement of equipment because these standards can be cited either as a baseline, with options and parameters procurer-defined, or as required specifications to which vendors must fully comply. The procurer is spared the task of providing a detailed technical specification for the interchange requirements and vendors are able to implement a relatively small number of standards as opposed to potentially a different set of interchange specifications to satisfy each purchase request.

Benefits specific to the use of document architecture (ODA or user-defined in SGML) and the related interchange formats include:

- a common technique for representing different types of information (e.g., text, facsimile, geometric, mosaic, audio) in the interchange format. (In the SGML standard, only text information has been standardized; however, the SGML-user may define a uniform way to represent non-SGML data, such as geometric graphics.)
- the capability to include different types of information in a single interchanged document; and
- provision of a structure which allows varying degrees of processability in the interchange format.

In short, the aims of both SGML and ODA are to achieve flexibility and compatibility for the user plus farsightedness in document interchange for both users and vendors.

- Title:** Comparison between the Main ODA and SGML Objectives and Proposal for Future Work ¹⁾
- Source:** *German Experts (U. Flasche, TU Berlin, A. Scheller, HMI Berlin)*
- Status:** The basics of this paper have been agreed, in principle, within DIN. Due to lack of time, it has not yet been discussed in sufficient detail to let it become a DIN comment.

In this paper a short comparison between the main objectives and features of the ODA (ISO DIS 8613) and SGML (ISO 8879) standards is given. Both standards are of fundamental importance because

- both standards are not simply regarding documents as flat strings of particular text information (e.g. characters or facsimile information) or as being in final layout, but are concerned with the "logical" structures of documents. Thus, they facilitate building convenient tools for distributed document handling which are important in many application areas, e.g. publishing, office applications, etc.
- both standards are supported by huge organizations which have made up their minds to use them for document exchange (such as the US Department of the Treasury, the AAP, etc., for SGML; the CCITT, the CEC, and European IT-industries for ODA).

Perhaps the question may arise whether *two* standards in the same area were really helpful and required, in particular if both have the same characteristics. For the ODA standard and the SGML standard this question is not applicable because their characteristics are significantly different: This statement will be discussed more precisely within the subsequent sections.

There are (at least) four different and important aspects, which may and must be taken as a fundamental basis when comparing the ODA and SGML models. These are:

¹⁾ This paper is an excerpt from:
ISOTEXT - An ODA/SGML WYSIWYG Editor/Formatter
S. Schindler, U. Flasche, C. Bormann, TU Berlin / TELES,
A. Scheller, HMI, to be submitted for publication in
"Informatik-Spektrum", Springer Verlag

1) *The Application Area Considered*

This aspect is concerned with the main objectives of the respective standard depending on the general application characteristics which have led to its development.

2) *The document architectures supported by the models*

The document architectural features of a document model arise from the facilities provided, within this model, for the structural composition/decomposition of documents, i.e. for defining the constituent parts (such as chapters, sections, paragraphs, footnotes, figures, pages, columns etc.) and the relations between them.

3) *The content architectures supported by the models*

Content architectures of a document model arise from the facilities provided, within this model, for dealing with different types of content information in a document.

4) *The interchange format(s) supported by the models*

This aspect of a document model is concerned with the features provided, within this model, for document interchange, i.e. with the ways of encoding all of the document information for sending it to some intended recipient of the document or fetching some document from its holder, resp.

1. Application Area

	SGML	ODA
Main Application Area Considered	<ul style="list-style-type: none"> • "Publishing environment", i.e. author creates document with logical markup, publisher performs all future processing on his own (possibly including distribution to "final" recipient) • In many cases closed application worlds • Document filing/retrieval objectives must be supported (document data bases) • Standardized layout characteristics not so important in most applications as no sender-driven layout process is to be performed by the recipient 	<ul style="list-style-type: none"> • "Office environment", i.e. (private) distribution of business letters, reports, etc. • In principle, documents must be sent to arbitrary recipients Document filing/retrieval objectives have been identified, the precise requirements of which are currently under discussion • Standardized layout characteristics for automatic reproduction at the recipient required (even after possible modification of the document at the recipient)

2. Document Architectures

	SGML	ODA
Universality	<ul style="list-style-type: none"> • Arbitrary semantics of a document structure • More than two "coexisting" structures possible • Thus "universal" 	<ul style="list-style-type: none"> • Distinction between logical and layout structures • At some particular point in time at most the logical and one layout structure of a document is used for processing • "Universal" for logical structures: arbitrary logical tree structures can be described (the possibility of defining additional logical attributes is depending on the flexibility of the "bindings" attribute) • Not "universal" for layout structures (predetined layout semantics)
Interchangeability between Arbitrary Communications Partners	<ul style="list-style-type: none"> • Documents, in general, not at all interpretable at arbitrary recipients • Semantics is currently fully dependent on particular application world • Envisaged registration of markup constructs will provide some predefined semantics 	<ul style="list-style-type: none"> • Predetined layout semantics (documents can be exchanged and presented at arbitrary recipients) • No predefined logical semantics (interpretation is dependent on application world) • It is hoped that registration features for (parts of) document classes will be provided in the future
Creation of Layout	<ul style="list-style-type: none"> • The embedding of arbitrary formatters is possible, but the choice will be performed for each application on its own. 	<ul style="list-style-type: none"> • A special "ODA" formatter (for a mapping of the ODA layout semantics to conventional formatters) is required, but it may then be used for all applications
Administration Information (Profile)	<ul style="list-style-type: none"> • No (standardized) administration attributes of a document as a whole 	<ul style="list-style-type: none"> • Set of standardized administration attributes for a document as a whole (author, security, keywords etc.)
Appropriateness of the Model	<ul style="list-style-type: none"> • Appropriate for describing arbitrary hierarchical document structures 	<ul style="list-style-type: none"> • Appropriate for describing hierarchical document structures for those applications for which the predetined layout semantics is reasonable

3. Content Architectures

	SGML	ODA
Universality	<ul style="list-style-type: none"> ● Content information of the chosen character set is directly expressable ● In principle, arbitrary content information is expressible via content notations 	<ul style="list-style-type: none"> ● Only predefined content architectures can be embedded. Currently: <ul style="list-style-type: none"> - character cont. arch. - raster graphics cont. arch. - geom. graphics cont. arch. Possible future extensions are e.g. <ul style="list-style-type: none"> - videotex - annotated voice, etc
Interchangeability between Arbitrary Communications Partners	<ul style="list-style-type: none"> ● In general, only the representation of the characters of the chosen character set is interpretable at an arbitrary recipient ● Content notations are totally application dependent 	<ul style="list-style-type: none"> ● The layout characteristics of all content information are predefined. Thus, all documents may be arbitrarily manipulated/presented at the recipient (if the corresponding content architecture is supported by the recipient).
Embedding of other Standards	<ul style="list-style-type: none"> ● The description of content information additional to the chosen character set might not always be derived from existing standards (danger of 'special solutions') 	<ul style="list-style-type: none"> ● Full integration of existing standards on content information (straight-forward embedding)

4. Interchange Formats

	SGML	ODA
Interchange Format	<ul style="list-style-type: none"> • SDIF 	<ul style="list-style-type: none"> • ODIF
Coding Model	<ul style="list-style-type: none"> • Sequence of descriptors in X.409/ASN.1 notation <ul style="list-style-type: none"> - chosen character set - "document body" including structural information - various kinds of additional "external" entities • Encoding of "document body" is character-oriented. • Structural information is separated from content information via special "escape" characters - Problem of escaping "escape" character. • Other coding schemes can only be added within the additional entities, i.e. outside of the document body 	<ul style="list-style-type: none"> • Sequence of descriptors and text units in X.409/ASN.1 notation <ul style="list-style-type: none"> - document profile - "document body" (consisting of generic layout descriptors, generic logical descriptors, generic text units, styles, specific layout descriptors, specific logical descriptors, specific text units) • Clear separation between content and structure information because of the TLV coding scheme of X.409/ASN.1 within the "document body". • Arbitrary content codings can easily be integrated within the "document body".
User Interface	<ul style="list-style-type: none"> • Notation of structural information is human-readable. • Notation of all content information may be kept human-readable if appropriate content notations are chosen • Thus, depending on the chosen character set and content notations, document input and output may be performed with some "old-fashioned" text system at a simple TTY (provided that the X.409/ASN.1 envelope or SDIF can somehow be created/interpreted) 	<ul style="list-style-type: none"> • Notation of structural information is not human-readable. • Notation of content information is not always human-readable • Thus, a special "ODA" text system is required for document input/output.

	SGML	ODA
Coding Efficiency	<p>A general statement on coding efficiency can hardly be made, because</p> <ul style="list-style-type: none"> • it depends, in particular, on the quantitative relation between markup information and contents information. • it depends on the attributes in use. • it depends on the length of GI names, attribute names, value names. • it depends on the applicable markup minimization which itself is depending on the possible structural composition of the markup elements. • it depends on the chosen content notations. • (depending on the chosen character set) it might depend on the quantitative relation of the content information of the character set in use to other content information. <p>Thus, it depends on the current document to be encoded. In most cases, SDIF might be slightly shorter than ODIF because</p> <ul style="list-style-type: none"> • the TLV overhead (resulting from the clear coding structure of X.409/ASN.1) is missing. • the clearly structured embedding of a large set of standardized attributes is missing. 	
Coder/Interpreter Efforts	<ul style="list-style-type: none"> • Coder/interpreter software must deal with <ul style="list-style-type: none"> - X.409/ASN.1, - special content information, - tag vs. content distinction (i.e. the syntactic conventions of the SGML language) 	<ul style="list-style-type: none"> • Coder/interpreter software must deal with <ul style="list-style-type: none"> - X.409/ASN.1, - special content information

5. Summary and Proposed Future Work

Summarizing the comparisons in the preceding sections, we see that the SGML standard is — quite intentionally — as "liberal" as possible, while the ODA standard provides — again quite intentionally — as many "regulations" as possible. Probably the most important reason for having these two different philosophies is that

- the ODA standard was designed by ISO — in particular by ECMA members — so as to achieve perfect cooperation with the other decisive international standardization bodies, in particular with the CCITT (and its decent body of recommendations so important for the European technical community) and also with other modern ISO standards, while
- the SGML standard was designed so as to be attractive and readily acceptable by the printing and publishing industries and their major customer organizations, such as e.g. the AAP.

Both philosophies are currently working out at their best. Thus it would be very unwise to try to amalgamate the SGML- and ODA-community by amalgamating both standards — because this would only disturb the two separate processes of penetration of these standards into the respective daily lives. Instead, text systems should be developed that are able to cooperate with both standards. Solving this technical problem is much easier than dealing with the political problems arising from melting both standards in a future ISO "super standard". Instead, two areas of future work concerning SGML/ODA "integration" can be envisaged:

5.1. ODA/SGML Comparison

It should be clear from the above comparisons and discussions that both standards have their advantages and disadvantages in a particular applications environment. In other words: There will be communities and their applications (mainly) using the one standard and other communities and applications (mainly) using the other standard.

Thus, a detailed comparison of the two models might be helpful for aiding potential applications in deciding which one of the two models may more perfectly fit to their needs. The short comparison sketched in this paper might be used as a basis for a more detailed comparison. This comparison may e.g. become a technical report of ISO.

5.2. Interworking Tools

Obviously, the SGML application community and the ODA application community will not be completely disjoint to each other. One of the reasons being that the decision, as to which standard has been accepted by a community, may have been more or less a matter of taste or just incidental. Consequently, documents should be exchangeable between members of both sets of communities. Therefore,

- the question should be discussed how to set up standards which would determine the conversion facilities for documents of both kinds, and
- the restrictions of the capabilities of such conversion facilities (due to features of documents which cannot be converted) should be investigated. Publishing these restrictions would leave the choice to all potential users of these standards, to use only those features of them which are supported in both of them.

Conversion ODA - SGML

The conversion of an ODA document to an SGML document is quite simple as far as the ODA standard is precise. Because of its universality, SGML may be used in a wide range of applications. Thus, the logical structures, layout structures and attributes permitted by the ODA model may be regarded as a particular (meta) application of SGML — by simply defining the ODA relevant attributes and structural decompositions by means of the SGML syntax. In addition, by using the concept of *coexisting structures*, SGML may also be used for concurrently describing both structures of the ODA model.

Several individual contributions are already available which describe a possible representation of the attributes of the ODA document structures and the ODA character content architectures in SGML notation. Similar contribution for conversion facilities for raster-graphics and geometric-graphics content informations should follow. In addition, a concept for expressing embedded control functions for character texts is still needed. Nevertheless, some problems with a pure notational conversion can be identified, in particular:

- A direct mapping of the ODA attribute "default value list" to SGML language features is not possible, because many levels of specifying default values are distinguished in ODA whereas in SGML only one default value can be specified in a markup declaration applicable for all occurrences of this markup element. Thus, the processing of default values in an ODA-conforming SGML document must be performed by some special application software.

- A general mapping of the ODA attribute "bindings" to SGML is difficult because of the complex syntax of this attribute and the grade of flexibility still left open in the ODA standard. Again, the conversion will have to be performed by some special application software.

Conversion SGML - ODA

The conversion of SGML documents to ODA documents must be performed either on an "SGML application basis" or as soon as the "structural elements" of SGML documents are standardized in an application independent way. As SGML has a very general scope, there are some features which cannot be directly converted into the ODA "environment" because the latter does not provide comparable features. This includes:

- processing instructions,
- marked sections,
- content notations which describe informations not provided by the ODA content architectures,
- entities in general (one has to decide for each concrete entity whether and how it can be converted).

15 July 1987

PLAN FOR CONFORMANCE TESTING
FOR DOD IMPLEMENTATION OF SGML

I. PURPOSE.

To define the plan for conformance testing of the DoD implementation of the Standard Generalized Markup Language (SGML). (Task 2.2.3.4.1)

II. BACKGROUND.

As an earlier CALS deliverable, NBS provided a program plan for conformance testing of standards supported by CALS. The deliverable stressed the importance of standards as well as conformance testing of these standards to ensure commercial availability of compatible off-the-shelf products. In order to show that standards are technically sound, they must be implemented and then proven to work. To prove the standard, these implementations must be tested. Later, the tests can prove product conformance when the standard is in use.

III. DISCUSSION.

The goal of the SGML conformance plan is to provide a structure and approach for conformance testing of the DoD implementation of the Standard Generalized Markup Language (SGML).

The SGML standard is a set of rules for defining applications such as a) the structure of document types (e.g., technical reports, technical manuals, training manuals, journals); b) the logical components of the document types independent of the format of those components (e.g., title, section headings, paragraphs); c) references to document contents that cannot be keyed from a keyboard or are external to the document (e.g., special characters, graphics, drawings); and d) specifications for database publishing systems. That is, SGML is a representation language for character text and it can be used for publishing in its broadest definition -- from single medium conventional publishing to multimedia database publishing. The user determines what components to identify within a document and describes those components in SGML.

Conformance testing of SGML thus far has focused on an integral part of an SGML system which is called an SGML parser. That is, there has been no attempt to test other components of an SGML system, such as SGML-sensitive editors or the SGML document interchange format (SDIF). In general terms, a parser is a program used to determine the underlying structure and content of some input object, such as a file or a document. In SGML terms, a parser is a program which checks that the tokens appearing in the input document occur in patterns that are permitted by the rules of SGML and that are permitted by the description given by

the document architect in the document type definition (DTD). The parser makes explicit the hierarchical structure of the incoming token stream by identifying which parts should be grouped together.

At least two types of testing are needed to prove conformance to the DoD implementation of SGML. These are 1) testing to prove that the SGML software (parser) works correctly and 2) testing to determine conformance to particular military standards (e.g., MIL-STD-38784B) used to create document type definitions (i.e., testing document structure). Additional tests are needed to prove that the interchange format of the SGML Document Interchange Format (SDIF) has been created, transmitted, and received correctly. As mentioned earlier, the conformance testing currently being done at NBS covers item one above. Future developments should include type 2 testing and SDIF testing.

NBS has been following a step-wise plan in the development of the type 1 tests. These steps are:

1. Stabilization of the standard. NBS began building an SGML parser over a year ago. The parser was implemented based on the Draft International Standard (DIS) version of the SGML standard. (At DIS, standards represent mature technology and little technical changes are expected to occur; this is the phase at which implementation of standards usually begins.) As the standard progressed through the standardization phases, changes to the parser necessarily followed. Now that SGML is an International Standard it is, for the most part, stable; however, there are some amendments which have recently been proposed. These amendments arose in part because of the NBS validation work; that is, ambiguities in the SGML specification were identified and the amendments are meant to correct these errors. This means that the parser will continue to be updated whenever there are amendments to the standard. NBS participates on the standards-making committees working on SGML to assure that Government requirements are met by the standard and that unnecessary changes to SGML will not be made.

2. Selection of features. Standards are the product of compromise. Because of this, there are often numerous optional features (resulting in varying functionality) which can be selected. One step in conformance testing is to agree on what will be tested --whether that be the full functionality of the standard including all optional features or whether a subset will be implemented and tested.

3. Implementation of standard. Once features and functionality have been selected and specified, the standard is implemented. The NBS implementation of SGML consists of all features required to conform to the International Standard and two optional

features, FORMAL and OMITTAG. (FORMAL indicates the use of a public identifier constructed according to SGML and OMITTAG indicates the use of markup minimization.)

4. Define test methodology. After determining what to test, a method (or how to test) for testing must be defined. The approach taken by NBS was addressed in a paper entitled, "SGML Parser Conformance Testing Methodology and Framework," which was an attachment to the 1986 CALS deliverable for text (Section I.3). Key points from the paper are summarized below.

- The goal of SGML implementations (parsers) should be portability of documents, i.e., the same document should result in the same interpretations when the document is parsed on different systems.

- Standardized test suites should be developed and these suites should evolve, rather than remain static, based on experience with SGML parsers.

- Test methods currently defined deal solely with the functional capacity of the SGML parser; other features of an SGML system, such as the user interface -- performance, parser design, and so on -- are not considered.

- The various levels of SGML implementations must be tested; therefore, the test suites will be structured to handle basic functionality as well as enhancements. Testing will begin with the most simple SGML document and will proceed logically through more complex documents up to the stated limits of the SGML parser being tested.

- In the development of test suites reliance on untested functions will be avoided, individual functions will be tested, and the number of tests will be minimized. In addition, the tests will be easy to use and the results easy to interpret.

5. Write test suite. In order to demonstrate that SGML parser software works correctly, NBS has developed an SGML validation test suite currently consisting of 470 tests. These tests cover those required features of SGML and the two optional features -- FORMAL and OMITTAG. Also, it should be noted that these tests address only the functional capacity of the SGML parser -- other aspects of an SGML system, such as user interface and performance, are not considered.

So far, these tests have been manually created, but the possibility of automatic or machine-generated tests for SGML documents should be explored.

6. Perform tests. NBS has executed the tests of the SGML validation test suite on our SGML parser. Using the results, the tests have been revised, as necessary. NBS has also made available early versions of the test suite so that other implementors can test their SGML software. So far, all testing, except the testing of the NBS SGML implementation, has occurred off-site by other implementors who execute the test suites on their own SGML parsers at their own sites rather than perform the testing at the NBS/ICST laboratory. NBS staff have assisted in areas of interpreting the SGML standard and have demonstrated the use of the test suite at several SGML validation workshops.

7. Report results. One of the objectives of the NBS/ICST Conformance Testing Plan (June 1987) is to remain "in sync" with Government and industry-practiced testing procedures for commercial products. Currently, thanks to the CALS initiative, NBS is not only "in sync," but has the lead in conformance testing for SGML software. NBS has hosted several SGML validation workshops where implementors have reviewed the SGML test suite devised by NBS and have productively interacted to improve these tests. (The test suite is expected to be completed by August 1987.) In addition, NBS has initiated ANSI and ISO projects to standardize conformance tests for SGML. The work of these parallel national and international projects is based on the output from the NBS-sponsored validation workshops.

8. Implement a Certification Program. A formal program needs to be in place to "certify" products for their conformance to SGML and SDIF. As the Conformance Testing Plan deliverable in June pointed out, a full program has to be implemented to accredit testing laboratories for certifying conformance to a standard and a mechanism has to be established for continued update and maintenance of the testing software.

IV. RECOMMENDATION.

The importance of conformance testing to assure compliance to a standard as well as to demonstrate workability of the standard cannot be over-emphasized. Conformance testing is an important part of the solution to realizing off-the-shelf solutions; therefore, it is recommended that DoD CALS continue to support the NBS SGML conformance testing effort which should include:

- defining document structure conformance tests - that is, type 2 testing to determine whether a document conforms to the structure prescribed by particular military standards, e.g., MIL-STD-38784B;
- defining (or review/acceptance of an already defined) testing methodology for SDIF tests;
- developing accreditation procedures to accredit

testing laboratories to test products implementing SGML and SDIF;

- identifying and accrediting testing laboratories to certify vendor products; and

- maintaining and modifying, when necessary, conformance tests for SGML and SDIF.

V. CONCLUSION.

Government agencies, and in particular the DoD, need technically sound national and international standards. Through conformance testing of implementations, the SGML standard can be demonstrated to work, these SGML implementations can be shown to conform to SGML, and soon products can be obtained which comply to the standard. Then the user can purchase off-the-shelf solutions to publishing application requirements.

RECOMMENDED DOD-M-(SGML) INSERTION

6.1.1.3 Guidelines for Tag Development.

These guidelines, for the creation of tags for use in developing new Document Type Definitions (DTD) or extending the Basic DTD, are based on the format of the tags that were created for the Basic DTD. Tags, in an SGML document, are descriptive markup or text that is added to a document to convey information about it. Tags are tightly coupled with the DTD, i.e., tags and the elements which they identify, must be integrated into the definition of the document type.

When new DTDs are developed for documents which can not be adequately described or handled under the Basic DTD, new tags may become necessary. New tags identifying new elements for an established DTD, however, will require changes to the DTD, to include the relationships of existing elements to the new elements. Changes to an existing Document Type Definition create a new Document Type Definition.

Those considering the development of new tags and elements, and new DTDs, should take note of the following:

- o The DoD Basic Tag Set in Appendix B is intended to be comprehensive across all potential SGML applications in DoD.
- o Development of new tags for special purpose applications may be necessary, but is discouraged.
- o Development and use of new tags requires the participation and coordination of the receiving data repository.
- o The Document Type Definition (Appendix A) and Output Specification (Appendix C) requirements must be considered when new tags are proposed.
- o Proposed additions to the DoD Basic Tag Set should be submitted to the Preparing Activity for DoD-M-SGML for consideration.
- o The tag-related optional features of the SGML language should not be used, as they are not included in DoD-M-SGML, and will not be recognized by SGML parsers being developed for SGML use.

Should it become necessary to create new tags and DTDs, some general rules to follow in creating tags are as follows:

- o Consult relevant military documents/standards to find appropriate terminology and definitions; do not invent new terminology, unless there is a need.

- o Tags should contain alphanumeric characters, with an alphabetic as the first character in the tag. Tags which have been developed for the Basic DTD follow this rule.
- o Tags should not contain special characters, e.g., slash, asterisk, and question mark. The hyphen and period are allowed, but are not recommended for use in tags. DCD has adopted the core concrete syntax in which the only non-alphanumeric characters permitted are the hyphen and period.
- o Tags should contain eight or fewer characters, as described in the reference quantity set.
- o New attributes should be considered for existing tags, if that will reduce the number of tags and meet your requirements.
- o Tags should convey meaning, if practical, to contribute to the general readability of the document.
- o New tags should not be invented for previously defined elements.

Purpose: The use of uniform tags will ease the interchange of documents between systems and can reduce the system requirements for storage of tags and elements if their lengths are kept to a manageable level.

The NBS FIPS-SGML Validation Suite
August 5, 1987

ABSTRACT

The Institute for Computer Sciences and Technology (ICST) of the National Bureau of Standards (NBS), through funds obtained from the Department of Defense (DOD) Computer Aided Logistics Support (CALs) initiative, has developed a validation suite which may be used as one tool in evaluating parsers which support the proposed Federal Information Processing Standard for the Standard Generalized Markup Language (FIPS-SGML) (Note: this proposed FIPS is currently under review). This product represents a deliverable to DOD and will also be placed in the public domain.

A fundamental goal of this validation suite is establishing a uniform validation technique to ensure that documents that conform to FIPS-SGML will be portable between the various applications of the Federal Government. This same validation suite may be useful to vendors and others who have interest in FIPS-SGML and who have need for document portability.

It must be understood that this validation suite is intended as one tool to be used in validating a FIPS-SGML parser - it does not in any way address the validation of a FIPS-SGML application.

1.0 PURPOSE

The purpose of this validation suite is consistent with the goal of FIPS-SGML - document portability. In the case of FIPS-SGML, document portability implies the ability to move a marked-up document from one machine to another without change and have that document (after parsing) produce identical results. This goal cannot be completely achieved unless parsers can be tested to determine their conformance to the relevant standard. The existence and acceptance of the NBS FIPS-SGML validation suite should lead to comparability and wide acceptance of test results produced by different examiners and ultimately to true portability of documents. We feel that this is fundamentally important to the Federal Government and offer this validation suite as one tool to use toward this goal.

Knowledgeable readers interpret the ISO 8879-1986 (the basis of FIPS-SGML) standard differently; this difference in understanding will likely result in parsers which do not produce identical results. By making available a validation suite which has been given intensive review it is hoped that these differences in understanding will be minimized and a higher level of document portability will result.

2.0 OVERVIEW

FIPS-SGML, the proposed Federal Information Processing Standard for the Standard Generalized Markup Language, is based on the Standard Generalized Markup Language (SGML) as described in International Standards Organization (ISO) 8879-1986. FIPS-SGML specifies a language for rigorously describing documents by the use of additional information interspersed amongst the text of the document. This additional information is called "markup" and serves to identify the logical elements of the document and may also specify processing functions to be performed. FIPS-SGML markup is characterized as descriptive rather than procedural and allows the document creator to identify those logical parts of the document that are important. This is done by associating a "tag" with each significant element of the document.

Subsequent to the marking up of the document, it must be processed to determine its underlying structure and its conformance to FIPS-SGML. The software which performs these functions is called a parser.

The NBS FIPS-SGML validation suite is a collection of test documents which may be used to test parsers which support FIPS-SGML. Since current technology does not support absolute validation of any complex piece of software, successful completion of the validation suite does not guarantee that a parser conforms to the FIPS-SGML (although a failure to successfully complete some part of the validation suite does show that the parser under test does not conform).

The validation suite is used to test parser conformance, not document conformance - this distinction is fundamentally important! Document conformance is basically a matter of syntax (in an SGML context, syntax is the set of rules governing the arrangement of the parts or elements of a document); if a document has been constructed according to the rules of FIPS-SGML, it is compliant. Furthermore, we can determine a document's conformance or nonconformance by inspection.

In contrast to document conformance which is described structurally, parser conformance is described functionally. The essential requirement is that a parser accept as input, any document which purports to conform to the FIPS-SGML standard, analyze its structure, and report errors (if any).

3.0 REFERENCES

3.1 Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), Document Number ISO 8879-1986(E).

3.2 Aho, Sethi, and Ullman, Compilers Principles, Techniques, and Tools, Addison-Wesley, Reading, Mass., 1986.

3.3 Beizer, Software Testing Techniques, Van Nostrand Reinhold, 1983.

3.4 Beizer, Software System Testing and Quality Assurance, Van Nostrand Reinhold, 1984.

3.5 Chandrasekaran and Radicchi, Computer Program Testing, North-Holland Publishing Co., 1981.

4.0 BACKGROUND

Work on the validation suite began in the summer of 1986 using the ISO Draft International Standard for the Standard Generalized Markup Language 8879 (DIS) as a basis. Preliminary work centered on understanding SGML; researching verification techniques; and considering approaches to automatically generating parts of the validation suite. At a workshop held at NBS in June 1986 representatives of industry, DOD, NBS, and the ISO and American National Standards Institute (ANSI) SGML committees began writing the validation suite and this work was continued throughout the summer.

In the fall of 1986, during an ANSI meeting at NBS, several members of the committee met with the NBS staff who were developing the validation suite to review the current effort. It was agreed that there was significant interest in the validation suite from the SGML user community and that another meeting should be held in the winter of 1987 to continue the review. In February 1987, at a meeting hosted by Hewlett-Packard, the validation suite was reviewed, expanded, and a common naming convention and format was identified.

The suggestions from the above meeting were used by NBS to refine and improve the validation suite and in June 1987 at another meeting hosted by Hewlett-Packard, a final review of the validation suite was made. The results of this review have led to the current validation suite.

Arrangements are being made with the National Technical Information Service (NTIS) to handle distribution of the validation suite. It is expected that NTIS will assume this function approximately January, 1988; an interim point of contact until that time is:

Fran Nielsen
National Bureau of Standards
Building 225, Room B266
Gaithersburg, Md. 20899

5.0 PROBLEMS IN DEVELOPING THE VALIDATION SUITE

Four main problems must be considered in creating a validation suite for FIPS-SGML parsers:

1. Access to the source code is not generally available.
2. The requirement for output from a FIPS-SGML parser is no different than that stated in ISO 8879-1986.
3. FIPS-SGML is not a programming language so it is not possible to have a parser check itself in any way.
4. There are ambiguities and omissions in parts of ISO 8879-1986.

The reality of item #1 does not allow us to consider structural techniques such as path testing, branch testing, code inspection, and formal proofs of correctness. Even if the source code were available, the above techniques would have to be redone for each parser. Instead, it is necessary to rely on functional testing techniques in which the parser is treated as a 'black box' to which input (in the form of conforming and nonconforming documents) is submitted, and from which the generated output (error or no error) is compared against the expected result.

Additional problems result from the definition of parser output (as defined in ISO 8879-1986): "A validating SGML parser shall find and report a reportable markup error if one exists, and shall not report an error when none exists."..."A report of an SGML markup error, including an optional report, shall state the nature and location of the error in sufficient detail to permit its correction." There are no output requirements for a parser beyond these and, consequently, some rather critical areas cannot be tested, e.g., proper substitution of default attribute values and proper handling of RS/RE's in some contexts. This limited output requirement also implies that we can only infer that the parser has processed the document correctly.

Item #3 requires that some process or person outside the parser evaluate the test results.

Item #4 is a significant problem. Because of ambiguous wording in many parts of ISO 8879-1986, knowledgeable readers do not always agree on the legality or illegality of certain constructions. NBS has resolved this problem by meeting with parser developers and members of the ISO and ANSI SGML committees to try to identify the most common interpretations of the wording of ISO 8879-1986 and these interpretations have been incorporated into the validation suite.

6.0 DESIGN CRITERIA

The basic goals of the design were:

1. Test every testable function of FIPS-SGML.
2. Keep the total number of tests to a minimum.
3. Test for proper handling of all relevant quantities in the reference quantity set.
4. Keep the individual tests as short as possible.
5. Test particularly those areas that are difficult to understand and/or difficult to implement.
6. Comment the tests sufficiently so that the intent and expected results are easily understood.

7.0 TESTING METHODOLOGY

As stated earlier, we must rely on functional testing of the parser under test, i.e., when conforming (valid) documents are submitted to the parser, we will consider the parser to have successfully completed the test if it does not report an error. Similarly, when nonconforming (invalid) documents are submitted to the parser we will consider the parse to have been successful if an error is reported "in sufficient detail to permit its correction."

8.0 DESCRIPTION OF THE VALIDATION SUITE

The validation suite is made up of 453 test documents, each of which is relatively short and primarily tests the parsers ability to process one construct of FIPS-SGML. On the release diskette, the test documents are grouped into logical areas known as subdirectories. These are named 'SECT06' - 'SECT13' and these are associated with the corresponding sections of ISO 8879-1986. Within these subdirectories are the files that correspond to the the test documents.

There is a standard naming convention for these files. If the filename begins with 'g' (good), then the document should be conforming, i.e., the parser should not report an error when it processes the document. If the filename begins with 'p' (prolog error), then the document is non-conforming and there is an error in the prolog. If the filename begins with 'i' (instance error), then the document is non-conforming and there is an error in the document instance. The last five characters of the filename reference the particular section of ISO 8879-1986 which is being tested. Leading and trailing zeros should be removed to find the associated section, e.g., if the last five characters are 09310 then the test applies to section 9.3.1.

There are some test documents (in subdirectory ERRATA) which are associated with ambiguous (knowledgeable persons interpret in different ways) parts of ISO 8879-1986; these tests represent NBS's interpretation of of the standard. Until ISO 8879-1986 is clarified, some developers will take different interpretations; it may be important for your application to understand how a parser will process these documents and this may be done by submitting these test documents to the parser.

There is another subdirectory on the release diskette named OPTIONAL. The tests in this subdirectory exercise optional error reporting features for FIPS-SGML parsers. Although this functionality is not a requirement of the parser as defined in ISO 8879-1986, some of these errors may result in significant (and unpredictable) errors during the processing of a document; these test documents may be used to determine whether or not a parser can recognize and report these types of errors.

The NBS FIPS-SGML Reference Parser

1.0 PURPOSE

The purpose of the FIPS-SGML reference parser is consistent with the goal of FIPS-SGML - document portability. In the case of FIPS-SGML, document portability implies the ability to move a marked-up source document from one machine to another without change and have that document (after parsing) produce identical results. This goal cannot be achieved unless the documents are, at a minimum, syntactically and semantically correct according to the rules of FIPS-SGML. The existence and acceptance of the NBS FIPS-SGML reference parser should lead to comparability and wide acceptance of documents produced by different sources and ultimately to true portability of documents. We feel that this is fundamentally important to the Federal Government and offer the reference parser as one tool to use toward this goal.

Knowledgeable readers interpret the ISO 8879-1986 standard (which is the basis for FIPS-SGML) differently; this difference in understanding will likely result in parsers that do not produce identical results. By making available a reference parser that has been given intensive review it is expected that these differences in understanding will be minimized and a higher level of document portability will result.

Finally, since the reference parser is distributed as source code with a well defined interface to user-developed applications, the parser could serve as a building block for various textual processing applications.

DISCLAIMER: Certain companies and specifications are mentioned in the text. In no such case does such identification imply recommendation or endorsement by the National Bureau of Standards.

2.0 OVERVIEW

FIPS-SGML, the proposed Federal Information Processing Standard for the Standard Generalized Markup Language, is based on the Standard Generalized Markup Language (SGML) as described in International Standards Organization (ISO) 8879-1986. FIPS-SGML specifies a language for rigorously describing documents by the use of additional information interspersed amongst the text of the document. This additional information is called "markup" and serves to identify the logical elements of the document and may also specify processing functions to be performed. FIPS-SGML markup is descriptive rather than procedural and allows the document creator to identify those logical parts of the document that are important. This is done by associating a "tag" with each significant element of the document.

Subsequent to the marking up of the document, it must be processed to determine its underlying structure and its conformance to FIPS-SGML. Using the terminology of ISO 8879-1986, the software that performs these functions is called a 'validating parser' (Note: ISO 8879-1986 also references a 'conforming parser'; all our references will be to the 'validating parser'). The parser referenced in ISO 8879-1986 has considerably more functionality than is usually associated with parsers in classical texts on language translation. For a more detailed discussion of this area, the user is referred to Appendix A or to the references listed in section 3.0.

The NBS FIPS-SGML reference parser is a set of programs that accepts as input some marked-up document. The parser processes the document, analyzing its structure and checking to see that it conforms to the rules of FIPS-SGML; if an error is detected, the processing terminates and an appropriate error message is displayed; if no error is detected a form of the document known as the 'Canonical Test Result' (CTR) is created (the CTR is described in Appendix C).

3.0 REFERENCES

3.1 Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), Document Number ISO 8879-1986(E).

3.2 Aho, Sethi, and Ullman, Compilers Principles, Techniques, and Tools, Addison Wesley, 1986.

3.3 Lewis, Rosenkrantz, and Stearns, Compiler Design Theory, Addison Wesley, 1978.

3.4 Schreiner and Friedman, Introduction to Compiler Construction with UNIX, Prentice-Hall, 1985.

3.5 Hopcroft and Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.

4.0 BACKGROUND

Work on the reference parser began in early fall of 1986, primarily to support the development of the FIPS-SGML validation suite. Due to significant changes between the Draft International Standard for SGML and the final International Standard, the work had to be restarted in November of 1986. As work progressed, there was sufficient interest from outside NBS to consider placing the parser in the public domain.

The parser has now been sent to several organizations and individuals and the feedback has been sufficiently favorable for NBS to generally make the parser available on request. Because the parser has been used repeatedly with the NBS FIPS-SGML validation suite, most of the obvious problems have been addressed - see APPENDIX B for a listing of known problems/limitations.

Arrangements are being made with the National Technical Information Service (NTIS) to handle distribution of the reference parser. It is expected that NTIS will assume this function approximately January, 1988; an interim point of contact until that time is:

Technology

Fran Nielsen
Institute for Computer Sciences and

National Bureau of Standards
Building 225, Room B266
Gaithersburg, Md. 20899

5.0 PROBLEMS IN DEVELOPING THE REFERENCE PARSER

The main problem in developing the reference parser was the readability of ISO 8879-1986. Many parts of the standard are ambiguously worded and knowledgeable readers disagree on the correct interpretation. These cases also presented problems in developing the validation suite; NBS staff made final decisions only after discussions and meetings with other developers and technical representatives of the ISO and ANSI SGML committees.

6.0 DESIGN GOALS

The basic goals of the design were:

1. Develop a simple to use parser that would compile and execute on UNIX (UNIX is a trademark of AT&T) and MS-DOS (MS-DOS is a trademark of Microsoft Corporation) based systems.
2. Produce a standalone package so that the user would not be required to have other utility programs.
3. Construct an application interface to the parser so that the parser could form the basis for textual processing applications.

7.0 BASIC DESIGN

{Note: It is not necessary to read this section to use the NBS FIPS-SGML reference parser. This information is made available for persons who need a more detailed understanding of the operation of the parser.}

Although it appears to operate as a single program, the parser is actually a collection of six executable programs. The following sections discuss the role of each program during the parse of a document.

[Note: Although there are six programs which make up the parser, the parser is not a multi-pass parser, i.e., the document is read only once. To minimize the size and complexity of the parser, the functionality was divided amongst the six programs. PARSE1.EXE reads to the document to the beginning of the document instance and writes several output files which are subsequently used by the remaining programs. PARSE1A.EXE, PARSE2A.EXE, and PARSE2B.EXE accept these files (not the source document) as input for further processing. PARSE3.EXE uses some of the intermediate files produced by PARSE1.EXE and 'fseeks' to the document instance of the source document to continue parsing. Upon normal termination, all work files are automatically deleted by the parser. must be invoked to parse a document, the document is actually read only once.

7.1 PARSE.EXE

This relatively simple program has the task of reading the command line, saving the information, and then invoking the other programs of the parser with the information they need to execute. To the typical user, this program is the parser and it is invoked by the following command:

```
parse filename
```

where filename is the name of the file containing the document to be parsed. PARSE.EXE checks to see that the document file exists and then, in sequence, invokes the remaining programs. If at any time an error is found, an appropriate message will be issued and the parsing process will cease.

7.2 PARSE1.EXE

This program reads and processes the SGML document entity up to (but not including) the beginning of the document instance set (see ISO 8879-1986 . The components of the prolog are checked (see APPENDIX B for known limitations) and if no errors are found a tree-like representation of the document (derived from the content models) is built. Other information created during the execution of this program is:

symbol table information
attribute value information
general and parameter entity information
inclusion and exclusion information

When the files containing this information have been built, PARSE1.EXE returns control to PARSE.EXE.

7.3 PARSE1A.EXE

This program traverses the tree-like representation of the document which was created by PARSE1.EXE and computes whether element tokens are contextually required (see ISO 8879-1986 sections 4.59, 4.60, 4.61, 7.3.1.1) in a content model. When this is done, PARSE1A.EXE returns control to PARSE.EXE.

7.4 PARSE2A.EXE

This program checks to see that each declared element has a path to the root element (the document type name as defined in section 11.1 of ISO 8879-1986) and also a path to a terminal data type (CDATA, RCDATA, EMPTY, #PCDATA). If either or both of these conditions are not met, a warning message is issued but parsing will continue. Upon completion of PARSE2A.EXE, control is returned to PARSE.EXE. The significance of an element not having a path to the root element is that there is no way for this element to legally occur in a document since the document must begin with the root element. The significance of an element not having a path to a terminal element is that the element may be improperly defined, i.e., once this element is encountered, it will may not be possible for parsing to continue.

7.5 PARSE2B.EXE

This program takes each content model and checks to see if it is ambiguous (see ISO 8879-1986 11.2.4.3). Ambiguous content models are displayed with an appropriate error message. If ambiguous content models are found, parsing will not continue to the next phase. Upon completion of PARSE2B.EXE, control is returned to PARSE.EXE.

7.6 PARSE3.EXE

This program performs the actual parsing of the document instance. It uses information from the previous programs to verify the syntax and semantics and provides information to the application interface as it encounters significant points in the document. If any error is encountered PARSE3.EXE issues an appropriate error message and stops.

8.0 ORGANIZATION OF THE RELEASE DISKETTE

The NBS FIPS-SGML reference parser is released on a 1.2 MB 5 1/4 inch floppy disk in MS-DOS format. All the executable programs may be found in the EXE directory off the root directory. A subdirectory, SOURCE, off the root directory contains the individual subdirectories of the source code for each of the programs described in section 7 above; the names of these subdirectories correspond to the program names (minus the suffix). In each of the source subdirectories there is a MAKE-like file ('make' is a utility that is associated with UNIX), which may be used to build the executable from the source. (You must substitute the name of your compiler in this file!)

Also, in subdirectory SOURCE, is a subdirectory named INCLUDE in which all the user written header (.h) files may be found - since these are shared amongst all the programs they are kept here for ease of maintenance. These header files are referenced by prefixing 'G:' to their name so if the user chooses to continue our convention of referencing them, he should use the SUBST command of MS-DOS as in:

```
SUBST G: A:\SOURCE\INCLUDE
```

APPENDIX A

A MORE DETAILED LOOK AT PARSING

It is important to understand that a parser does not print a document, build a database, etc. These functions are performed by applications that may interface to the parser, but it is absolutely incorrect to state that they are done by the parser. In the simplest of terms, a validating parser analyzes the structure of a document and checks the document for conformance to FIPS-SGML.

ISO 8879-1986 (4.2.8.5) defines an SGML parser to be "A program (or portion of a program or a combination of programs) that recognizes markup in conforming SGML documents." It further states (15.4.1) "A validating SGML parser shall find and report a reportable markup error if one exists, and shall not report an error when none exists."

As stated earlier, the functionality of a parser as defined in ISO 8879-1986 is somewhat more than most of the classical texts assign to it; to differentiate an SGML parser as defined in ISO 8879-1986 from the more common definition, we will use the term 'SGML parser' to represent the former and 'parser' for the latter. An SGML parser may be considered to have at least three components: a lexical scanner, a parser, and a semantic analyzer.

The lexical scanner (or analyzer) has the task of reading the input data stream and breaking it up into meaningful units called tokens. For instance in analyzing the following declaration:

```
<!ELEMENT x CDATA>
```

the lexical analyzer might recognize seven meaning units:

	1.	'<!'	<Markup Declaration Open
		(mdo)>	
	2.	'ELEMENT'	<SYNTACTIC KEYWORD>
	3.	' '	<Parameter Separator
(ps)>			
	4.	'x'	<NAME>
	5.	' '	<Parameter Separator
(ps)>			
	6.	'CDATA'	<NAME>
	7.	'>'	<Markup Declaration Close
(mdc)>			

The output of the lexical analyzer becomes the input to the parser which performs syntax analysis (in the context of SGML, syntax is the set of rules governing the arrangement of the parts or elements of a document). More formally, the parser accepts the string of tokens from the lexical analyzer and checks to see that they form a legal sequence according to the rules of FIPS-SGML. Finally, the semantic analyzer may check semantics and

perform any associated processing which must be done, e.g., to associate name 'x' with an index to a symbol table, etc.

Certainly, the boundaries between lexical analyzer, parser, and semantic analyzer, are somewhat arbitrary and different implementors will place functions in whichever piece is convenient for their system.

A FIPS-SGML parser can check a document to see that it conforms syntactically and semantically to the rules of FIPS-SGML but it cannot check that the information in the document is valid, it cannot format a document, it cannot build a database, etc. These functions must be done by a process (or person) outside the parser!!

APPENDIX B

KNOWN LIMITATIONS OF THE NBS REFERENCE PARSER

The following is a list of known limitations of the NBS reference parser. Most, if not all, of these areas will seldom if ever prove a problem to most users. As time and interest permit, NBS may make enhancements to the parser to remove some or all of these. (Note: The following descriptions are necessarily somewhat technical.)

1. The optional (per ISO 8879-1986, 15.4.1) reporting of an exclusion that could change a groups required or optional status in a model is not implemented.
2. The optional (per ISO 8879-1986, 15.4.1) reporting of a failure to observe a capacity limit is not implemented.
3. The optional (per ISO 8879-1986, 15.4.1) reporting of an error in the SGML declaration is not implemented.
4. The optional (per ISO 8879-1986, 15.4.1) reporting of a formal public identifier error is not implemented.
5. In processing an Attribute Definition List Declaration, the normalized length of the attribute specification list may be miscalculated.
6. Warning messages may be erroneously given that an element lacks a path to the root element and/or lacks a path to a terminal element when, in fact, these paths do exist by way of exceptions. (In these cases, the messages should be ignored.)

Appendix C

The Canonical Test Result Format of a Document

[Note: The Canonical Test Result (CTR) format of a document has been suggested as a standardized output for a validating SGML parser. The CTR is not yet a part of any standard and is subject to change - the NBS parser implements the CTR as it is described in this appendix. The following explanation of the CTR was taken from a report following a workshop on SGML parser validation.]

The Canonical Test Result can be used to evaluate tests of valid SGML documents. The CTR was designed to indicate that all SGML constructs have been correctly parsed. It is an output format that is intended to be compared to a pre-determined correct answer and is not intended to be subsequently parsed. Hence there is no need to indicate within the CTR when characters that might be interpreted by a human reader as delimiter characters happen to occur as data.

The CTR uses the following indicator characters:

- [- Markup start indicator
-] - Markup end character
- | - Line break indicator
- ? - Processing instruction indicator
- / - Element end indicator
- # - Non-SGML character indicator
- & - External entity indicator

In the CTR, every markup start indicator that does not occur as data is placed in the first column of a line and every markup end indicator that does not occur as data is the last character on the line containing it. Whenever a line would be longer than the maximum line-length quantity, which is nominally 60, the line is broken so that the next character is placed in column 1 of the following line. To avoid line-trailing white-space, whenever a line would otherwise end with a space or tab, a line break indicator is placed in the following (nominally, the 61st) position.

In contexts where the Reference Concrete Syntax does not distinguish between uppercase and lowercase letters in name tokens, the name always appears in uppercase form in the CTR.

The CTR reflects the interpretation of markup within the document instance as follows:

At the start of every element encountered in the document instance, whether or not the element's start-tag is minimized, the CTR outputs a markup start indicator followed by the element's generic identifier. The element's attributes are listed on the following lines, in the order in which they are declared, one attribute per line. Each attribute name begins in column 2 and is followed by a single space, an equals sign, and another space. Following the second space, if the attribute's value is impliable, is the character string #IMPLIED. If the attribute's value is known, either by applying a default, or because it was explicitly entered in the element's start-tag, the value is entered, surrounded by quotation marks. Note that, since the CTR is not parsed, a quotation mark may appear with the quoted string. Further note that if the attribute value would extend beyond the line-length quantity, the value is entered over successive lines. If the last character on one line would be a white-space character, it is followed by the line break indicator. After the last attribute, a line containing only the markup end indicator in column 1 is generated. For example, the start of an occurrence of element A00-G1 might be indicated in the CTR as:

```
[A00-G1
 A      0      0      -      A      1      =
"1.....10.....20.....30.....40.....|
.....60"
 A00-A2 = "A00-V1"
 A00-A3 = #IMPLIED
]
```

The end of every element is indicated with a line in the CTR containing a markup start indicator, followed by the element's generic identifier (GI), followed by a markup end indicator. This line is even included for elements that may not have an end-tag because their declared content is EMPTY. An example follows:

```
[/A00-G1]
```

Every record end (RE) that is treated as data appears as a line in the CTR containing only a markup start indicator followed by a markup end indicator:

```
[ ]
```

A processing instruction is preceded by a markup start indicator in column 1 and followed by a markup end indicator:

```
[?Text of Processing Instruction]
```

A non-SGML character is indicated by its character number preceded by a markup start indicator and non-SGML character indicator and followed by a markup end indicator:

[#5]

A reference to an external entity is indicated by the entity name preceded by a markup start indicator and the external entity indicator and followed by a markup end indicator:

[&A00-E1]

Note that except for external entity references and references to non-SGML characters, no indication is made in the CTR that short references, entity references, or character references have occurred. Instead, the result of processing the reference appears in the CTR. Similarly, marked sections are processed and comment declarations are discarded.

Appendix D

Interfacing to the NBS FIPS-SGML Reference Parser

[Note: The information in this appendix is not required to use the parser as it is distributed; it is intended to be used by persons (with knowledge of the C programming language) having a need to build an SGML application based on the NBS parser.]

There is a provision in the NBS parser to interface to a user-developed application; this interface exists in the file SEMANTIC.C in the subdirectory PARSE3 of the release diskette. A listing of this file is given below:

```
#include <stdio.h>
#include "g:semantic.h"
/*-----*/
/*-----*/
/*-----*/
/*
This function is called by the parser at 'significant' points in
the document. These points are identified by the value of the
variable 'code' which will be:

1. END_TAG_NAME - when an end tag is encountered, e.g.,
  </para>; str1 will point to a null terminated string
  containing the element name.
2. TAG_NAME - when a start tag is encountered, e.g., <para>;
  this will be followed by 0 or more calls to this function
  with code = TAG_ATTR; str1 will point to a null terminated
  string containing the element name.
3. TAG_ATTR - this value is passed for each attribute associated
  with a start tag; str1 will point to a null terminated string
  containing the attribute name, str2 will point to a null
  terminated string containing the attribute value.
4. TAG_END - this value is passed when there are no more
  attributes associated with a start tag.
5. DATA_STG - this value is passed to give text (not markup)
  to the application; str1 points to a null terminated
  character string.
6. PROC_INST - this value is passed to give the content of a
  processing instruction to the application; str1 points to
  a null terminated character string.
*/
void semantics(code, str1, str2)
int code;
char *str1, *str2;
{
    switch(code) {
    case END_TAG_NAME:
#ifdef APPL_DEBUG
        printf("*** END_TAG_NAME='%s' ***\n", str1);
#endif
    }
}
```

```

#endif
    break;
    case TAG_NAME:
#ifdef APPL_DEBUG
    printf("*** TAG_NAME='%s' ***\n",str1);
#endif
    break;
    case TAG_ATTR:
#ifdef APPL_DEBUG
    printf("*** TAG_ATTR='%s'='%s' ***\n",str1,str2);
#endif
    break;
    case TAG_END:
#ifdef APPL_DEBUG
    printf("*** TAG_END ***\n");
#endif
    break;
    case DATA_STG:
#ifdef APPL_DEBUG
    printf("*** DATA_STG='%s' ***\n",str1);
#endif
    break;
    case PROC_INST:
#ifdef APPL_DEBUG
    printf("*** PROC_INST='%s' ***\n",str1);
#endif
    break;
}
return;
}

```

As is apparent from the code and comments, this function is called at significant points in the document and is passed information which may be useful to an application. The user should simply replace the code for each case statement with his/her own code. If there is any uncertainty as to how this interface functions, the 'printf' statements may be enabled by defining APPL_DEBUG, recompiling, and then parsing a simple document. The resulting output should clarify how and when this function is called.

USING SGML
APPLICATION GUIDELINES

ABSTRACT

The Institute for Computer Sciences and Technology (ICST) of the National Bureau of Standards (NBS), through funds obtained from the Department of Defense (DOD) Computer Aided Logistics Support (CALs) initiative, has developed a reference parser which may be used as one tool in evaluating documents that conform to the Standard Generalized Markup Language (FIPS-SGML) (Note: this proposed FIPS is currently under review). This product represents a deliverable to DOD and will also be placed in the public domain.

A secondary goal of this effort is to provide a simple to use product that can be used to learn FIPS-SGML; in the same way that a programming language is best learned by writing programs and compiling them, FIPS-SGML may be learned by creating documents and submitting them to the parser. This document is intended to further that second goal by providing an introduction to the concepts and uses of SGML, as well as providing examples of the usage of the NBS reference parser.

I. Introduction

During the last decade, there has been a proliferation of computer equipment and software for creating, editing, revising, processing, and transmitting textual information. Unfortunately, users -- like you -- have discovered through experience that it is often difficult or impossible to exchange information among various systems. As users become more aware of these problems, they have also become insistent on compatibility in order to avoid being trapped in single-solution dead ends and to avoid paying the price of either converting documents from one product's format to another or rekeying documents.

Several efforts are underway to remedy this situation. In the area of text interchange, one such effort has been the standardization of a markup language to facilitate the exchange of textual information among computer-processing systems.

The remainder of Section I provides a brief description of markup, the Standard Generalized Markup Language (SGML), and an SGML parser. Section II discusses how you can use SGML, including how to write Document Type Definitions (DTDs). In order to show that standards are technically sound, they must be implemented; Section III explains the NBS Reference Implementation of an SGML parser and discusses its use by showing examples. These implementations must be tested. Later, the tests can demonstrate product conformance when the standard is in use. NBS has developed a conformance plan and validation test suite to evaluate SGML parsers. Section IV describes this validation suite.

What is Markup ?

Markup refers to instructions annotating a manuscript. There are two kinds of markup -- descriptive markup and procedural markup. Descriptive markup identifies the structure of a document without regard to the document's ultimate presentation whereas procedural markup describes the appearance of the document.

In computerized text-processing systems, markup is often embedded within the subject text in a source file. Markup languages have made processing easier and provide for flexibility in output by separating a document's text from its instructions. The instructions may indicate style for features (such as titles or

Note:

This document is intended to provide an accessible overview of SGML and how it may be applied to the problem of document production and interchange. It is not intended as a substitute for the SGML specification. For more detail, please refer to the SGML standard and the documents mentioned in the references.

lists,) or the features themselves. Indication of features is generalized markup, while indication of style is procedural markup.

What is SGML?

The Standard Generalized Markup Language (SGML) ISO-8879-1986 is an International Standard for describing document structures by identifying the elements, or features, of the document. SGML, a methodology for creating tagging schemes, defines the rules for document description and markup; however, SGML is not a set of tags for document elements nor a template for any particular document type.

The SGML standard specifies a language for document representations. The standard provides for:

- a) the structure of document types (e.g., technical reports, technical manuals, training manuals, journals);
- b) the logical components of the document types independent of the format of those components (e.g., title, section headings, paragraphs); and
- c) references to document contents that cannot be keyed from a keyboard or are external to the document (e.g., special characters, graphics, drawings.)

The user determines what components to identify within a document and describes those components in SGML.

SGML includes a syntax for descriptive markup of document elements, provides for arbitrary data content, includes entity references for referring to separate non-SGML data, and includes processing instructions.

II. Using SGML

The major parts of an SGML document include the Document Type Definition (DTD) and document instance, or document element. This section describes DTDs and document elements.

The DTD provides an unambiguous grammar for the logical composition of a class of documents. The grammar has a hierarchical structure and provides descriptive information about the elements and the relationships between those elements. No formatting information is explicitly provided in the DTD. The DTD must conform to the rules set by the standard in terms of unambiguous content, limits on numbers of tokens, and depth of nesting, among others. (Note that the limits can be user defined. SGML defines only the default values as the reference quantity set. Adherence to these values insures maximum document portability.)

The following is a sample DTD for a simple memo:

```
<!doctype memo [  
<!element memo - - (mheader,body) >  
<!element mheader 0 0 (to, from, date?, re)>  
<!element (to | from) 0 0 (name) >  
<!element name - 0 CDATA >  
<!element (date | re) - 0 CDATA >  
<!element body 0 0 (par | list)* >  
<!element par 0 0 (emph ; regular)* >  
<!element emph - 0 CDATA >  
<!element regular - 0 CDATA >  
<!element list 0 0 (ltag | ltext)* >  
<!element ltag - 0 CDATA >  
<!element ltext - 0 (#pcdata) >  
>]
```

This DTD displays the hierarchical structure of an SGML document. A memo contains a memo header (mheader) and a memo body (body). A memo header includes "to," "from," and "subject" elements, and optionally, a date element. Through the memo header, the memo includes all these elements as well.

Some of the key constructs are also displayed in this simple DTD. The '?' character in the memo header indicates that the "date" element is optional. The "|" character in the definition of the "to" and "from" elements indicates a logical "OR." The asterisk in the body definition indicates an occurrence property - in this case, the elements "par" or "list" can occur one or more times. A plus sign would have indicated zero or one occurrence. (The absence of an occurrence indicator implies exactly one occurrence.) These constructs provide sufficient power and flexibility to describe extremely complex documents.

The elements will be distinguished in the text by tags. An SGML tag is simply an ASCII character string, beginning and ending with special character sequences, and containing the name of the element. The tags are used to indicate the beginning and ending of that portion of the document. Each occurrence of an element begins with a start tag and ends with an end tag. Whatever falls between these tags, if anything, is the element's "content." The content of an element can be raw data and/or other elements, as specified in the DTD. Each element declaration ends with the content specification for the element. So, the content of an element must be an element header followed by an element body.

The document instance is the actual tagged document. The document is a hierarchically related series of elements, as defined in the DTD. The document instance must begin and end with tags for a valid document type. So, the document instance is essentially an element whose generic identifier is a document type name referenced in the DTD.

Please note that the users may not have to explicitly tag the document themselves. Applications may "hide" SGML from the user, just as a text editor hides the ASCII or EBCDIC encoding from users. This is desirable, but some applications will probably require that users know the tagging structures.

In the case of our sample DTD, "memo" is the document element. The content of the document element includes the remainder of the elements in the document. The elements must all conform to the SGML standard and the document instance they comprise must conform to the associated DTD.

A sample document instance for a short memo conforming to the sample DTD is given below:

```
<memo><mheader>
<to><name>Visitor</name></to>
<from><name>NBS Electronic Publishing Lab Project
Team</name></from>
<re>SGML Semantic Processing</re>
</mheader>
<body><par><regular>
In conjunction with the NBS Electronic Publishing Lab's support of
format standards development, prototype SGML Semantic Generators
are being developed on the systems in the lab.
</regular></par></body></memo>
```

The basic label constructs are as follows:

- <memo> is a start tag for element memo
- </memo> is the end tag for an occurrence of element memo
- everything in between <memo> and </memo> is the content

Writing the DTD

As, mentioned earlier, the SGML document instance must be written to a DTD; that is, the document must adhere to the structure defined by the DTD.

Elements included as part of a document might be titles, headings, figures, figure captions, paragraphs, and words. The elements which can occur in an SGML document are described, and their relationships specified, in the Document Type Definition. This description will consist of a content specification, and optionally, an attribute list. Attributes are simply characteristic qualities other than type or content. An example would be a security attribute for a memo, with legal values of "Unclassified", "SECRET", and "TOP SECRET." Elements can occur in a document only according to the rules of the DTD, and may only have those attributes defined in the DTD.

The SGML standard defines a document as a set of hierarchical structures. A manual, for example, could be composed of front matter, a series of one or more chapters, optional appendices, and an index. A chapter could be composed of sections, which are composed of text, and optionally, figures, tables, lists, and so on.

Many different documents can be written to the specifications of a single document type.

There are several steps in writing a DTD: selecting a document or groups of documents (e.g., technical manuals) for which the DTD will be written, identifying the elements and entities of that document, laying out the structure of the document, developing the DTD and testing it. These steps are briefly described below.

Developing a DTD requires the analysis of a set of documents with similar content to determine the structural elements and their relationships. The documents selected should be representative of many potential documents of this type. Alternatively, if no prototype documents are available, it can be developed based upon a concept of what the particular documents should contain. From this analysis, the parts or elements of the document must be

identified along with their attributes. Entities that may be referenced within the documents must be identified, and the structures that may occur in the particular document type defined.

There is a shorthand markup in SGML that indicates that a substitution will occur. Such markup is an entity reference. Markup that defines the entity is the entity declaration. When a name, portion of text, chapter in a book, title, or whatever, has been previously defined or will be used several times throughout a document, it can be defined as a separate entity using a short identifier or abbreviation. At the point in the document where it is needed, the entity can be referenced. The entity will be substituted for the reference when the document is processed. Keystrokes can be substantially reduced through the use of entities. An entity reference is of the form:

&SGML; where the (&) and (;) are the beginning and end delimiters. The SGML reference is to the entity: Standard Generalized Markup Language.

You must declare user-defined entities in the DTD. Public entities would allow the utilization of public DTDs, specialized graphics, and character sets not supported directly on the keyboard (such as Greek or Latin.)

An analysis of the documents will help to identify the significant elements, their attributes, and their relationships. This means that you will decide the hierarchy of the document elements where they can occur, how often they can occur, and in what combinations they can occur. The parts of a document are identified by the following types of markup:

- * tags - descriptive; delimit structural elements
- * entity reference - denotes substitution;
the reference is replaced by an entity
- * declarations - define the rules, i.e., environment
- * processing instructions - system specific;
controls the document processing

Each of the types of markup is separated from the document's text by strings of characters, called delimiters. The delimiters are different for each type of markup, and can be redefined by the DTD's author, but such redefinitions could reduce document portability, since some SGML applications may adhere strictly to these limitations.

SGML does not specify what tags (i.e., names) should be used to identify the elements of a document. The names of elements are defined by the author of the DTD. It is suggested that they correspond to the actual structures and elements the document will contain. The only constraint imposed is a maximum tag length. The

default value in the Reference Quantity Set is eight characters. Again, this number may be adjusted to suit the needs of the user, but it may reduce document portability. Eight characters will usually be enough to allow mnemonic names. For example, names for elements within a manual might include the following: title for title; para for paragraph; list for list; litem, for list item; chaphead, for chapter heading, etc. While paragraph and chapter heading had to be abbreviated, they are still recognizable.

Working With The Document Instance

An SGML parser may be used to 1) check the validity of the DTD and 2) check the document element for conformance to the DTD. The parser may be a discrete step in a multi-step process, or hidden as part of an application. It may be one pass in a translation process to prepare an SGML document for output.

Error handling is not rigorously defined in the SGML standard; but usually, any errors in the document element will result in no output or incorrect output. Errors in the DTD will certainly require attention before further progress can be made.

An SGML document is less likely to contain problems in document structure than a document prepared with a less rigorous system, but review of content will, of course, remain necessary.

Using the Document Instance

The driving concepts of SGML are the ideas of flexibility and ease of document interchange between document processing systems. Since the markup does not include system dependent procedural markup, transparent document translation and transmission is possible. This means greater flexibility in printing (or displaying), storing, and moving the document.

Printing the document will require replacing tags with the processing commands of the target output device (i.e, local commands), as well as some formatting of the elements' data content. For example, the output device may be a CRT, a dot matrix printer, or a laser printer producing a camera ready final copy. The commands to drive the device and format the document will differ for each device.

SGML does not specify the way in which a document is stored, or moved. The SGML Document Interchange Format Standard (SDIF) is a standard (ISO 9069-1987(E)) intended to address the problem of transferring and storing an SGML document as a single data stream, while allowing the user to reconstitute the document in its original form. SDIF is intended solely for the exchange of conforming SGML documents among conforming SGML systems. The interchange may take place via Open Systems Interconnection communications or by the exchange of storage media, or other

means.

The document may be stored in any or all of these forms: SGML, SDIF, a page description language, or a local format (such as that generated by a text editor.) Note that storage in any of the page description languages or local formats will result in some loss of information, since the information implied by the tags will be lost. Storing a document in device dependent form may be desirable for frequent or interactive access; however, those forms will not necessarily be revisable. Retention of SDIF or SGML forms is usually desirable.

Additional SGML Constructs

There are many additional constructs which are included in SGML to make it easier to use or provide additional power. They address several topics, including: minimizing tags; using non-standard character sets; multiple versions of documents; and explicit specification of hierarchical position of an element. These features are not necessary to produce most English language documents, but may be helpful in many situations.

Minimization

Minimization allows the omission of redundant tags. For example, the example document element could be code as follows:

```
<memo><mheader>
<to><name>Visitor</to>
<from><name>NBS Electronic Publishing Lab Project Team</from>
<re>SGML Semantic Processing</re>
<body><par><regular>
In conjunction with the NBS Electronic Publishing Lab's support of
format standards development, prototype SGML Semantic Generators
are being developed on the systems in the lab.
</par></memo>
```

without any ambiguity. The "name" element must end before the end of the "to" element or "from" element, so the </name> tag is redundant information. The element "mheader" must end before the body begins, so the </mheader> is unnecessary, and so on.

The element declarations in the DTD have two previously unexplained entries after the generic identifier which appear as hyphens or 'O's. The first entry specifies minimization properties for start tags for that element; the second for the end tag. The hyphen indicates the tag is required, 'O' indicates that the tag is optional.

For example:

```
<!element mheader O O (to,from,date?,re) >
```

The element "mheader" has optional start and end tags.

```
<!element (date | re) - O CDATA >
```

The elements "date" and "re" require start tags, but the end tags may be omitted. Note that tags omission may be permitted only if the document element will be unambiguous without them.

External entities

An SGML application may have an entity manager that will allow the user access to resources supplied by the system external to the application. External entities allow access to external storage objects such as files and libraries without introducing system dependencies to the body of the document. The nature and syntax of the entity statement will depend on the individual entity manager, which will perform the task of converting the entity into a system address.

External entities may be either private or public. A public entity is one known beyond the context of an individual document or system environment. Public entities could be used to provide consistent DTDs for an entire organization. They could also be used to provide a shared character set for specialized terms and letters (such as Greek letters for scientific applications.)

Character references

In cases where it is inconvenient or impossible to enter a character directly, a technique called character reference is available to enter that character indirectly. For example, assume that some terminals at your site have no hyphen. These terminals are attached to a system using a character set where a hyphen is character number 45. Then, the entity declaration

```
<!ENTITY hyphen "&#45;">
```

would allow your users to enter "‐" into the data content of an element to insert that character. (The user could also enter "-" directly, but that would make the document incorrect on any new system where a hyphen was not character 45.)

Ranked elements

The ranked elements feature allows the DTD's author to create classes of elements with inherent nesting information. The elements each have a unique generic identifier of the form pn, where p is the (common) rank stem and n is a (unique) number called the rank suffix. The DTD entries would be as follows:

```

<!Element p 1      - O (#PCDATA, p2*) >
<!Element p 2      - O (#PCDATA, p3*) >
<!Element p 3      - O (#PCDATA, p4*) >
<!Element p 4      - O (#PCDATA) >

```

When the rank feature is used, only the rank stem must be entered in the tag. The rank suffix will be maintained until the next fully specified generic identifier. Of course, if a new level will be utilized, the rank suffix must be supplied. For example, suppose you wanted to input a list, and had the additional element declaration

```

<!Element p      - - (p1*) >

```

with p the element defined as a list. Then the list could be tagged in several fashions, including those given below.

<u>minimally tagged</u>	<u>fully tagged</u>	<u>formatted output</u>
<p><p1>abc	<p><p1>abc</p1>	1. abc
<p>def	<p1>def	2. def
<p2>geh	<p2>geh</p2>	a. geh
<p>ijk	<p2>ijk</p2></p1>	b. ijk
<p1>lmn</p>	<p1>lmn</p1></p>	3. lmn

This is a convenient way to specify rank explicitly. Note, however, that explicit notation is usually not necessary if the SGML translator takes advantage of the semantic information available in the document markup. The formatted output could have easily been produced without ranked elements.

Marked sections

Documents with multiple versions, or documents with processing instructions specific to certain systems can be handled by a technique called marked sections. Marked sections facilitate that maintenance of such documents, allowing one document to address all versions of the problem. Marked section declarations usually involve defining the different version names to represent the SGML keywords INCLUDE or IGNORE, depending on which version is desired. Changing the version would then require only exchanging which keywords the entities were defined as. The marked section appears as:

```

<![ %version; [elements or processing instructions or...] ]>

```

and would be processed by the parser if version was defined as INCLUDE and skipped if it was IGNORE.

III. SGML Parsers

What is an SGML parser?

According to ISO-8879 1986, an SGML parser is "a program (or portion of a program or a combination of programs) that recognizes markup in conforming SGML documents." A parser with only this level of functionality is known as a conforming parser; ISO-8879 defines another type of parser known as a validating parser requiring that the "parser shall find and report a reportable markup error if one exists, and shall not report an error when none exists."

Certainly a validating parser must also recognize markup in a conforming document (and therefore is also a conforming parser) but a validating parser must also be able to handle documents that contain errors according to the SGML standard (non-conforming documents). Since we are primarily interested in helping you understand how to create correct (conforming) SGML documents and since validating parsers are used to check documents for conformance, all references to a parser in this guideline should be taken to mean validating parser unless specified otherwise. In the simplest of terms, a validating parser will read the document and determine whether or not it conforms to ISO 8879.

Although different implementors will approach the task of parsing in their own ways, it is reasonable to consider the parsing of the document prolog as distinct from the parsing of the document instance. The document prolog may generally be considered to be the document type definition and contains various combinations of declarations (element declarations, entity declarations, etc), references (general and parameter entity references, character references), comments, and processing instructions. Some of the tasks that must be done while processing the prolog are:

1. Reading each declaration and verifying that it is correct according to the rule that defines it in ISO 8879.
2. Resolving entity references as they are encountered.
3. Checking processing instructions to see that they only occur at specified locations and that their length is within the allowed range.

As the document type definition is being parsed, it will also be necessary to save various information for later - the names of the elements and the order in which they may appear, the legal and default values for attributes, the definitions for entity references, etc.

After the necessary information has been generated from the document type definition, processing (parsing) of the document instance can begin. During this phase the parser will encounter

constructs such as start tags, end tags, attributes, entity references, character references, processing instructions, etc. Using the information generated earlier, the parser ensures that start and end tags occur at their expected positions, that the attribute values are legal, etc. Another important function that must be performed at this time is the recognition of implied markup (omitted tags and attribute values).

Although an SGML document must contain both a prolog and a document instance, it is not a requirement that they be in the same file. In addition, it is not a requirement that a single program parse both of them. Depending on the parser that you have available, you might invoke one program to parse the prolog and then another to parse the document instance or you might invoke a single program to cause all parts of the document to be parsed. In any event, the output from the parser will include some indication of whether or not an error occurred and if so, some information to help you locate it.

What is the NBS SGML reference parser?

The NBS SGML reference parser (hereafter called the parser) is a collection of C source programs which, when compiled and linked, will function as a validating parser. The parser supports documents coded in the core concrete syntax (see ISO 8879-1986 section 14) with the OMITTAG and FORMAL features (see ISO 8879-1986 section 13.5.1 and section 13.5.3). This corresponds to the SGML declaration specified in DOD-M-SGML. Although developed on generic PC's, the parser has no known hardware or operating system dependencies and should be portable to any system which offers a C compiler.

This parser was developed and placed in the public domain to provide an easy to use tool for users to check their SGML documents for conformance. The typical scenario would be that a document would be created with some word processor or text editor and this document would be input to the parser. The user would then review the output of the parser; if an error(s) was indicated, the user should correct the error and repeat the process again. This should be done until the document parses without error.

The NBS SGML reference parser and its use are more fully described in "The NBS FIPS-SGML Reference Parser."

Sample use of the NBS reference parser

As an example, let's see how the NBS reference parser could be used to correct the following document. (Note: there are intentional errors in this document!)

```
<!DOCTYPE MEMO [  
<!ELEMENT MEMO - O( (DATE, (ORIGIN? & DESTINATION), SUBJECT, BODY,  
COPIES) )>  
<!ELEMENT (DATE,ORIGIN,DESTINATION,SUBJECT) - O(#PCDATA)>  
<!ELEMENT BODY - O(#PCDATA)>  
<!ELEMENT PARA - O (#PCDATA)>  
<!ELEMENT COPIES - O(LIST)>  
<!ELEMENT LIST - O(LISTHEAD, LISTITEM+)>  
<!ELEMENT (LISTHEAD,LISTITEM) - O(#PCDATA)>  
<!ATTLIST MEMO security (UC, CON, SECRET, TS) #REQUIRED>  
>
```

```
<MEMO >  
<DATE>28July, 1987  
<ORIGIN>NBS-ICST  
<SUBJECT>SGML User's Guide  
<BODY>  
<PARA>  
  This is the first paragraph of the user's guide.  
<PARA>  
  asertoi9uawenktyt60asretoiu the sUMof 2plusTWO=9  
<COPIES>  
<LIST>  
<LISTHEAD>  
COPIES to:  
<LISTIEM>  
  NBS  
<LISTITEM>  
  DOD  
<LISTITEM>  
  CALS  
</MEMO>
```

Suppose this document is in a file called 'MEMO' and we want to verify that it conforms to the rules of SGML. First, we parse the document and examine the messages from the parser - this is done by entering:

parse MEMO

from the command line.

The output from the parser would look something like this:

```
PARSE1.EXE test
<!DOCTYPE MEMO [
```

```
----- E R R O R -----
in document: MEMO
name too long in group
Current Declaration =
<!ELEMENT MEMO - O (DATE,(ORIGIN?&DESTINATI
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

The line containing 'PARSE1.EXE test' indicates which part of the parser is active (for a more detailed discussion of the components of the NBS parser, please refer to ???). In this case, the document prolog is being parsed. Now look at the error message - it states that a name is too long and further shows which declaration was being processed. Referring back to the original document, we see that the identifier DESTINATION appears in the content model. Using DOD-M-SGML, we are limited to a set of values known as the reference quantity set shown as Figure 6 (page 52) in ISO 8879-1986. This quantity set allows at most eight characters in a name (as indicated by the quantity NAMELEN in Figure 6 of ISO 8879-1986). We must therefore change the name 'DESTINATION' to something eight or fewer characters in length. This must be done in both the document type definition and the document instance.

Note that the text of the document may have names of arbitrary length - the eight character limit applies only to names, name tokens, numbers, etc. which are recognized by the parser. Suppose we change DESTINATION to DEST everywhere it appears in the markup and try again.

```
PARSE1.EXE test
<!DOCTYPE MEMO [
<!ELEMENT MEMO - O (DATE, (ORIGIN?&DEST), SUBJECT, BODY, COPIES)>
<!ELEMENT (DATE,ORIGIN,DEST,SUBJECT) - O (#PCDATA)>
<!ELEMENT BODY - O (#PCDATA)>
<!ELEMENT PARA - O (#PCDATA)>
<!ELEMENT COPIES - O (LIST)>
<!ELEMENT LIST - O (LISTHEAD,LISTITEM+)>
<!ELEMENT (LISTHEAD,LISTITEM) - O (#PCDATA)>
<!ATTLIST MEMO
      SECURITY (UC,CON,SECRET,TS) #REQUIRED>
```

```
total elements = 11
total attributes = 1
total entities = 0
```

```
PARSE1A.EXE
PARSE2A.EXE
WARNING: element PARA may not have path to root element
PARSE2B.EXE
PARSE3.EXE test -P382
<MEMO >
Error: REQUIRED or CURRENT attribute not specified 'SECURITY'.
```

This time we went much further; looking at the output from the parser we see no errors or warnings from PARSE1.EXE and also none from PARSE1A.EXE. We do see that the part of the parser known as PARSE2A.EXE issued a warning indicating that the element PARA may not have a path to the root element. The root element in our example is MEMO and this message is advising us that the parser cannot find a path from MEMO to PARA. The rules of SGML do not define this to be an error so only a warning is issued; in reality, the user should check to see that this is not a problem. In the case of MEMO, what this means is that we have declared an element, PARA, which is not in the content model of any other element. Let's assume that we really meant for BODY to be made up of things called PARA and change the element declaration for BODY accordingly.

There is also an error message from PARSE3.EXE which complains about an attribute called SECURITY. In reviewing the document type definition, we see that there is an ATTLIST declaration for MEMO and that SECURITY is a required attribute, i.e., it must be specified by the user and cannot be defaulted. Looking at the allowable values for security, we choose UC and add the attribute information to the start tag for MEMO. Now, let's try to parse the document again.

```

PARSE1.EXE test
<!DOCTYPE MEMO [
<!ELEMENT MEMO - O (DATE, (ORIGIN?&DEST), SUBJECT, BODY, COPIES)>
<!ELEMENT (DATE, ORIGIN, DEST, SUBJECT) - O (#PCDATA)>
<!ELEMENT BODY - O (PARA)>
<!ELEMENT PARA - O (#PCDATA)>
<!ELEMENT COPIES - O (LIST)>
<!ELEMENT LIST - O (LISTHEAD, LISTITEM+)>
<!ELEMENT (LISTHEAD, LISTITEM) - O (#PCDATA)>
<!ATTLIST MEMO
      SECURITY (UC, CON, SECRET, TS) #REQUIRED>

```

```

total elements = 11
total attributes = 1
total entities = 0

```

```

PARSE1A.EXE
PARSE2A.EXE
PARSE2B.EXE
PARSE3.EXE test -P379
<MEMO security='UC'>
<DATE>28July, 1987
<ORIGIN>NBS-ICST
<SUBJECT>S
Error: Invalid tag, last opened tag 'MEMO'.

```

Another error, this time we have just seen the start tag for 'SUBJECT' - the question is to determine what has gone wrong. Referring back to the document type definition, we see that the content model for MEMO is:

(DATE, (ORIGIN?&DEST), SUBJECT, BODY, COPIES)

and in looking at the document we see that elements DATE and ORIGIN have occurred and the element SUBJECT is being processed. What about DEST? This element was not optional but we have forgotten to include it in the document instance; let's put DEST in the document instance and try again.

```
PARSE1.EXE test
<!DOCTYPE MEMO [
<!ELEMENT MEMO - O (DATE, (ORIGIN?&DEST), SUBJECT, BODY, COPIES) >
<!ELEMENT (DATE,ORIGIN,DEST,SUBJECT) - O (#PCDATA)>
<!ELEMENT BODY - O (PARA)>
<!ELEMENT PARA - O (#PCDATA)>
<!ELEMENT COPIES - O (LIST)>
<!ELEMENT LIST - O (LISTHEAD,LISTITEM+)>
<!ELEMENT (LISTHEAD,LISTITEM) - O (#PCDATA)>
<!ATTLIST MEMO
      SECURITY (UC,CON,SECRET,TS) #REQUIRED>
```

```
total elements = 11
total attributes = 1
total entities = 0
```

```
PARSE1A.EXE
PARSE2A.EXE
PARSE2B.EXE
PARSE3.EXE test -P379
<MEMO security='UC'>
<DATE>28July, 1987
<dest>
CALS document repertoire
<ORIGIN>NBS-ICST
<SUBJECT>SGML User's Guide
<BODY>
<PARA>
  This is the first paragraph of the user's guide.
<PARA>
```

Error: Invalid tag, last opened tag 'MEMO'.

Well, we did get further this time but again we have an error. A start tag for PARA had just been seen - by going back to the document type definition, we see that the element BODY has a content model of PARA with no occurrence indicator, i.e., only one occurrence of PARA is allowed within BODY. Since BODY can have multiple paragraphs, we will associate a 'plus' occurrence indicator with PARA in the content model for BODY and try again.

```

PARSE1.EXE test
<!DOCTYPE MEMO [
<!ELEMENT MEMO - O (DATE, (ORIGIN?&DEST), SUBJECT, BODY, COPIES) >
<!ELEMENT (DATE, ORIGIN, DEST, SUBJECT) - O (#PCDATA) >
<!ELEMENT BODY - O (PARA)+ >
<!ELEMENT PARA - O (#PCDATA) >
<!ELEMENT COPIES - O (LIST) >
<!ELEMENT LIST - O (LISTHEAD, LISTITEM+) >
<!ELEMENT (LISTHEAD, LISTITEM) - O (#PCDATA) >
<!ATTLIST MEMO
      SECURITY (UC, CON, SECRET, TS) #REQUIRED >

```

```

total elements = 11
total attributes = 1
total entities = 0

```

```

PARSE1A.EXE
PARSE2A.EXE
PARSE2B.EXE
PARSE3.EXE test -P380
<MEMO security='UC'>
<DATE>28July, 1987
<dest>
CALS document repertoire
<ORIGIN>NBS-ICST
<SUBJECT>SGML User's Guide
<BODY>
<PARA>
  This is the first paragraph of the user's guide.
<PARA>
  asertoi9uawenktyt60asretoiu the sUMof 2plusTWO=9
<COPIES>
<LIST>
<LISTHEAD>
COPIES to:
<LISTIEM>
Error: Unknown generic identifier 'LISTIEM'.

```

Once more we have progressed farther into the document. This time the parser is reporting an unknown generic identifier called LISTIEM. Looking at this carefully, we see that we have simply miskeyed the identifier and it should be LISTITEM. After changing this in the document, we give it to the parser again.

```
PARSE1.EXE test
<!DOCTYPE MEMO [
<!ELEMENT MEMO - O (DATE, (ORIGIN?&DEST), SUBJECT, BODY, COPIES)>
<!ELEMENT (DATE, ORIGIN, DEST, SUBJECT) - O (#PCDATA)>
<!ELEMENT BODY - O (PARA)+>
<!ELEMENT PARA - O (#PCDATA)>
<!ELEMENT COPIES - O (LIST)>
<!ELEMENT LIST - O (LISTHEAD, LISTITEM+)>
<!ELEMENT (LISTHEAD, LISTITEM) - O (#PCDATA)>
<!ATTLIST MEMO
      SECURITY (UC, CON, SECRET, TS) #REQUIRED>
```

```
total elements = 11
total attributes = 1
total entities = 0
```

```
PARSE1A.EXE
PARSE2A.EXE
PARSE2B.EXE
PARSE3.EXE test -P380
<MEMO security='UC'>
<DATE>28July, 1987
<dest>
CALC document repertoire
<ORIGIN>NBS-ICST
<SUBJECT>SGML User's Guide
<BODY>
<PARA>
  This is the first paragraph of the user's guide.
<PARA>
  asertoI9uawenktyt60asretoiu the sUMof 2plusTWO=9
<COPIES>
<LIST>
<LISTHEAD>
COPIES to:
<LISTITEM>
  NBS
<LISTITEM>
  DOD
<LISTITEM>
  CALS
</MEMO>
```

Normal program termination.

This time the parser has processed the complete document and has found that the document is conforming, i.e., it follows the rules of ISO 8879-1986. The above example was chosen to contain many errors so you could see the iterative process of editing and parsing; this may not be typical of real documents that you create and there are authoring tools known as context sensitive or syntax directed editors which will ensure that the documents you create

are conforming. It should also illustrate that a parser can only identify SGML errors - a parser cannot tell us whether the actual content of the document is correct or not. In the case of our test document, the content of the second PARA is just gibberish. Some person or process outside the parser must review the documents for these types of problems.

IV. The NBS SGML Validation Suite

The NBS SGML validation suite is a collection of 453 documents which may be used as one tool in evaluating SGML validating parsers as described in section III. By submitting these documents to a parser and comparing the actual to the expected results, you may develop a better understanding of how closely the parser conforms to ISO 8879-1986.

Apart from using the validation suite as one tool in parser evaluation, the validation suite can also be used as a learning tool. By examining the documents that make up the validation suite (these are well commented), you may develop a better idea of what is permissible and what is not in SGML.

Although the validation suite may be used as one tool in evaluating SGML parsers, there are other important factors that should be considered; some of these are:

product support - no matter which parser you choose, if you use it often enough you are likely to have questions which can best be answered by the people who developed the parser. Depending on how important this is to you, you may want to choose a product backed by a good user support policy.

ease of use - is the parser easy to use, does it provide good error messages, etc.

cost - parsers range in price from free to several thousand dollars.

other required software - some parsers are self contained, i.e., they do not require you to have other programs such as compilers, linkers, etc. Others may require that you have these programs available on your system - be sure to ask the vendor if support software is required.

other features - some parsers are coupled to applications that drive publishing systems or intelligent editors. These may or may not be of interest to you.

support for SGML functions - most parsers today do not support all the functions that are possible in SGML; if you need these, seek a parser that supports them.

availability of source code - if you plan to add an application to the parser, this may be less difficult if you have access to the source code; if you are primarily non-technical and do not plan to add to the

functionality of the parser, this issue is irrelevant.

processing speed - how quickly a parser can process a document.

How to use the NBS validation suite

On the release diskette, the test documents are grouped into logical areas known as subdirectories. These are named 'SECT06' - 'SECT13' and these are associated with the corresponding sections of ISO 8879-1986. Within these subdirectories are the files that correspond to the test documents.

There is a standard naming convention for these files. If the filename begins with 'g', then the document is conforming, i.e., the parser should not report an error when it processes the document. If the filename begins with 'p', then the document is non-conforming and there is an error in the prolog. If the filename begins with 'i', then the document is non-conforming and there is an error in the document instance. The last five characters of the filename reference the particular section of ISO 8879-1986 which is being tested. You should remove leading and trailing zeros to find the section, e.g., if the last five characters are 09310 then the test applies to section 9.3.1.

There are some test documents in which not everyone will agree that the document is conforming or non-conforming. ISO 8879-1986 is vague in some areas and these tests represent NBS's interpretation of SGML. Tests that are known to be somewhat controversial are in the subdirectory named ERRATA on the release diskette.

There is another subdirectory on the release diskette named OPTIONAL. The test documents in this subdirectory may be used to see if a particular parser implements some of the optional error reports specified in section 15.4.1 of ISO 8879-1986.

V. Glossary of terms

The definitions given in this section are those found in the International Standard or in reports providing tutorials for the SGML standard.

attribute: provide supplementary information about an element that qualifies the generic identifier

attribute declaration: specifies the relationship between an entity and its attribute(s)

descriptive markup: identifies the structure of a document without regard to the document's ultimate presentation

document element: a logical part of a document delimited by a start-tag and a matching end-tag

document type declaration: a markup declaration that contains the formal specification of a document type definition.

document type definition: a statement that specifies the elements that may be included in a document of the defined type and the contexts in which the elements may occur

element declaration: a markup declaration that contains the formal specification of the part of an element type definition that deals with the content and markup minimization

end-tag: descriptive markup that identifies the end of an element

entity declaration: the association of a name with an entity so that the entity can be referenced (e.g., SGML to represent Standard Generalized Markup Language)

generalized markup: descriptive markup imparting no formatting information

marked section declaration: a markup declaration that identifies a marked section and specifies how it is to be treated.

markup: information added to the data content of a document to enable its constituent elements to be distinguished from one another and/or information to aid in processing a document (i.e., processing instructions)

markup minimization: techniques to reduce the number of keystrokes when indicating the markup of a document

ODIF: the Office Document Interchange Format (ISO 8613 Part 5)

procedural markup: describes the appearance of the document

processing instruction: markup consisting of system-specific data that controls how a document is to be processed

SDIF: the SGML Document Interchange Format (ISO 9071)

SGML: the Standard Generalized Markup Language (ISO 8879-1986)

start-tag: descriptive markup that identifies the start of an element and specifies its generic identifier and attributes

tag: a string of characters identifying an element type

VI. References

1. Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML); ISO 8879-1986(e) October 1986.
2. MARKUP - External Specification for Document Authors. Simon Bate, Lynne A. Price. 25 June, 1985, Tandem Computers Incorporated, External Version.
3. The Standard Generalized Markup Language (SGML): Guidelines for Editors and Publishers. Joan Smith. British National Bibliography Research Fund Report 26. 1987
4. The Standard Generalized Markup Language (SGML): Guidelines for Authors. Joan Smith. British National Bibliography Research Fund Report 27. 1987
5. Information Processing Systems - Open Systems Interconnection - Reference Model 1S7498 1984.
6. Information Processing - SGML Support Facilities - SGML Document Interchange Format (SDIF) Draft International Standard (DIS) 9069. May 1987.
7. "The NBS FIPS-SGML Reference Parser," to be distributed, August 1987.
8. Aho, Sethi, and Ullman, Compilers Principles, Techniques, and Tools, Addison Wesley, 1986.
9. Lewis, Rosenkrantz, and Stearns, Compiler Design Theory, Addison Wesley, 1978.
10. Schreiner and Friedman, Introduction to Compiler Construction with UNIX, Prentice-Hall, 1985.
11. Hopcroft and Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.
12. Beizer, Software Testing Techniques, Van Nostrand Reinhold, 1983.
13. Beizer, Software System Testing and Quality Assurance, Van Nostrand Reinhold, 1984.
14. Chandrasekaran and Radicchi, Computer Program Testing, North-Holland Publishing Co., 1981.

TITLE: ODA/ODIF Implementation Agreement
A DOCUMENT APPLICATION PROFILE

STATUS: Submitted to NBS OSI Implementor's Workshop for
inclusion in "on-going" Agreements Document at the
October 9, 1987 Workshop Plenary.

EDITOR: Fran Nielsen, NBS

DATE: October 9, 1987

ODA/ODIF Implementation Agreement

- 12.1 Introduction
 - 12.1.1 References
- 12.2 Scope and Field of Application
- 12.3 Status
- 12.4 Errata
- 12.5 Assumptions
 - 12.5.1 Conformance
- 12.6 Document Architecture
 - 12.6.1 Characteristics of This DAP
 - 12.6.2 Notation and Conventions
 - 12.6.2.1 Notation
 - 12.6.2.2 Superclasses
 - 12.6.2.3 Superclass Expressions
 - 12.6.2.4 Use of Bindings Expressions
 - 12.6.2.5 Object Class Identifiers
 - 12.6.3 Logical Components
 - 12.6.3.1 Logdoc
 - 12.6.3.2 Passage
 - 12.6.3.3 Section
 - 12.6.3.4 Number
 - 12.6.3.5 Title
 - 12.6.3.6 Text&Refs
 - 12.6.3.7 FNote
 - 12.6.3.8 FNBody
 - 12.6.3.9 Figure
 - 12.6.3.10 Text
 - 12.6.3.11 Graphic
 - 12.6.3.12 Reference
 - 12.6.3.13 Ref
 - 12.6.3.14 Raster
 - 12.6.3.15 Geometric
 - 12.6.4 Layout Components
 - 12.6.4.1 Laydoc
 - 12.6.4.2 Pageset
 - 12.6.4.3 Page
 - 12.6.4.4 RPage
 - 12.6.4.5 VPage
 - 12.6.4.6 Header
 - 12.6.4.7 Footer
 - 12.6.4.8 BodyFR
 - 12.6.4.9 FrameA
 - 12.6.4.10 FrameB
 - 12.6.4.11 FrameC
 - 12.6.4.12 Framed
 - 12.6.4.13 FrameE
 - 12.6.4.14 FrameF
 - 12.6.4.15 FrameG
 - 12.6.4.16 FrameH
 - 12.6.4.17 Block
 - 12.6.5 Attributes
 - 12.6.5.1 Attribute Applicability

- 12.6.5.2 Attribute Values
- 12.7 Content Architecture
 - 12.7.1 Character Content
 - 12.7.1.1 Character Formatted
 - 12.7.1.2 Character Processable
 - 12.7.1.3 Character Formatted Processable
 - 12.7.2 Raster Graphics Content
 - 12.7.2.1 Raster Graphics Formatted
 - 12.7.2.2 Raster Graphics Processable
 - 12.7.3 Geometric Graphics Content
- 12.8 Document Profile
 - 12.8.1 Presence of Document Constituents
 - 12.8.2 Document Characteristics
 - 12.8.3 Document Management Attributes
- 12.9 Interchange Format
- 12.10 Relationship to Other Document Application Profiles
 - 12.10.1 SPAG
 - 12.10.2 CCITT
 - 12.10.3 TOP
 - 12.10.4 POSI
- 12.11 Examples
- 12.12 List of Tables
- 12.13 List of Figures

12. DOCUMENT ARCHITECTURE AND INTERCHANGE FORMAT

12.1 INTRODUCTION

Section 12 defines an Implementors' Agreement based on Office Document Architecture (ODA) and Interchange Format, as defined in ISO DIS 8613 and provides detailed specification for the implementor. Such an agreement is termed a Document Application Profile according to ISO DIS 8613.

ISO DIS 8613 has seven parts:

Part 1 of the DIS gives an introduction to the standard as a whole and provides a description of the general principles of ODA;

Part 2 defines the document structure model and the document processing model;

Part 4 defines the document profile and its use;

Part 5 defines the interchange formats;

Part 6 defines the character content architectures;

Part 7 defines the raster graphics content architectures;

Part 8 defines the geometric graphics content architectures.

12.1.1 References

The following documents are referenced in the statement of the agreements relating to Office Document Architecture.

- [1] Information processing : Text and Office Systems Office Document Architecture (ODA) and Interchange Format Part 1: Introduction and General Principles - ISO/DIS 8613/1 June 1987
- [2] Information processing : Text and Office Systems Office Document Architecture (ODA) and Interchange Format Part 2: Document Structures - ISO/DIS 8613/2 June 1987
- [3] Information processing : Text and Office Systems Office Document Architecture (ODA) and Interchange Format Part 4: Document Profile - ISO/DIS 8613/4 June 1987
- [4] Information processing : Text and Office Systems Office Document Architecture (ODA) and Interchange Format Part 5: Office Document Interchange Format - ISO/DIS 8613/5

June 1987

- [5] Information processing : Text and Office Systems Office Document Architecture (ODA) and Interchange Format Part 6: Character Content Architectures - ISO/DIS 8613/6 June 1987
- [6] Information processing : Text and Office Systems Office Document Architecture (ODA) and Interchange Format Part 7: Raster Graphics Content Architectures - ISO/DIS 8613/7 June 1987
- [7] Information processing : Text and Office Systems Office Document Architecture (ODA) and Interchange Format Part 8: Geometric Graphics Content Architectures - ISO/DIS 8613/8 June 1987

12.2 SCOPE AND FIELD OF APPLICATION

This is the definition of a document application profile suitable for interchanging documents in processable form. This document application profile is defined in accordance with ISO DIS 8613.

The document application profile is intended for transfer of mixed-mode documents between currently existing document processing systems; that is, this DAP is intended for documents potentially containing character text, raster graphics, and geometric graphics. Thus, the document application profile is appropriate for document processing systems that are designed to use non-impact printers but not necessarily designed to use ODA. These are typified by "desk top publishing" systems.

The documents addressed by this document application profile range from simple memos to highly structured technical reports or articles.

This document application profile defines features covering the document characteristics, character content layout and imaging, character repertoire, graphics content, and document management.

12.3 STATUS

This is the third working draft of the ODA/ODIF implementation agreements, October 1987.

12.4 ERRATA

None; in the future this section will contain corrections and clarifications to this version of the agreements.

12.5 ASSUMPTIONS

12.5.1 Conformance to This Document Application Profile

12.5.1.1 ODIF Datastream Conformance

This document application profile (DAP) separates the "permissible" range of values for attributes, as specified in ISO 8613, into "basic" and "non-basic" values. Basic values are a subset of the permissible values that constitute the "basic set." All other permissible values are considered to be non-basic values.

This document application profile defines a conforming "basic datastream" to be a valid ODIF encoding of a document that contains only constituents as defined in this DAP and contains no attributes or values outside of the basic set. A conforming "basic interpreter" is a product that correctly interprets any conforming basic datastream and may have more capability as well. A conforming "basic generator" is a product that produces only conforming basic datastreams, or can reliably be directed to function in a mode of producing only conforming basic datastreams.

12.6 DOCUMENT ARCHITECTURE

12.6.1 Characteristics supported by this document application profile

The following sections describe the logical and layout features that can be represented in documents conforming to this document application profile. The features are described in terms that are typical of the user-perceived capabilities and semantics found in current document processors. The features are grouped into logical features and layout features in order to relate them to their ODA representation.

Documents conforming to this document application profile can contain character text, geometric graphics and/or raster graphics contents.

12.6.1.1 Logical characteristics

Logical document structure

The logical structure of documents comprise sections, passages, paragraphs, figures and footnotes. Sections can be nested and automatic section numbering is provided for.

The logical structure of a document conforming to this document application profile consists of a hierarchy of logical objects. The following is an example of a generic document logical structure derived from this document application profile:

```
Document
  Passage (s)
    Paragraph
      Initial text
      Footnote
        Footnote reference
        Footnote body
      Continued text
      Figure
      Continued text
      Figure
      Section level 1
        Section number level 1
        Section title
        Passage
          Paragraph
          Figure
        Section level 2
        ....
```

Document structure elements

1 Document

A document is composed of a sequence of passages.

For example, separate passages may included (a) the contents to be placed on the title page of a report (b) the body of the report and (c) the contents to be placed in appendices.

2 Passage

A passage consists of any logical sequence of sections, paragraphs and/or figures that can be regarded as an entity for reading or for layout presentation.

A table is a particular case of a passage.

A single paragraph or a single figure is a simple case of a passage.

The layout of passages is described in 12.6.4.

3 Section

A section has an automatic section number which precedes any other contents and serves to identify the section for human comprehension.

The contents of a section may begin with a section title starting on the same line as the section number.

A section may contain one or more passages which may be followed by a sequence of sections within the enclosing section.

The document originator may define different classes of sections having in common some presentation features and/or some layout features. For example, the document originator may define a class of sections which always begin on a new page, and another class of sections which are laid out using a special left or right margin offset.

The layout of sections is described in 12.6.4.

Automatic section numbering

An automatically generated section number consists of a series of numbers separated by instances of an arbitrary specified character string content. It is equal to the automatically generated section number (if any) of the enclosing section followed by a single index number to uniquely identify the

section.

Index numbers are generated sequentially within any section. The method of numbering for each level (e.g., the 4th number) must be the same throughout the document. It may be any of:

- a) Arabic numerals
- b) Upper/lower case letters
- c) Upper/lower case Roman numerals

5 Paragraph

A paragraph is a contiguous amount of character text in the intended reading order.

A paragraph contains zero, one or more embedded footnote references. Multiple consecutive footnote references, without intervening text, are permitted.

A paragraph contains zero, one or more embedded figure associations. Multiple consecutive figure associations, without intervening text, are permitted.

A paragraph may comprise a number of character sequences concatenated together, for example if the character sequences were separately derived or generated.

The document originator may define different classes of paragraphs having in common some presentation features and/or in some layout features. For example, the document originator may define classes of paragraphs for "abstract", "standard paragraph", "hint" or "summary".

The layout of paragraphs is described in 12.6.4.

6 Figure

A figure is an amount of geometric graphics or raster graphics content designed to occupy a rectangular area.

One or more paragraphs can be associated with a figure in order to provide captions or notes.

The layout figures is described in 12.6.4.

7 Footnote

A footnote consists of a footnote reference and a footnote body.

The footnote body is a contiguous amount of text that can be read out of sequence from the paragraph containing a reference to it.

The layout of footnotes is described in 12.6.4.

8 Footnote reference

A footnote reference may have an automatically generated label or one supplied by the user. If the label is automatically generated then the label may be represented by Arabic numerals, upper or lower case Roman numerals, or upper or lower case letters.

12.6.1.2 Layout characteristics

Document Layout Structure

The following is an example of a generic document layout structure derived from this document application profile:

```
Document
  Page set
    Page
      Header area
      Body area
        Single frame
        Multiple columns
        Individual frame(s)
        Mixed set of frames
      Footer area
```

....

Document layout structure elements

1 Document

A document consists of a sequence of one or more page sets.

2 Page set

The pages within a page set all have the same dimensions and orientation (landscape or portrait) but may differ in layout and/or content of the header and footer areas.

There may be an optional first page of one particular page layout and this may be followed by either of the following:

- a) Repeated pages with the same layout
- b) Repeated pages designed for alternating recto and verso layout

3 Page layout

This document application profile supports page dimensions up to the assured reproduction areas of the ISO A4 nominal page size, in portrait and landscape orientation.

A page layout consists of:

- a) An optional header area that is reserved for header contents
- b) A single body area
- c) An optional footer area that is reserved for footer contents

Each of these areas must be totally contained within the assured reproduction area of the nominal page dimensions and must not overlap with the other areas.

Particular header and footer contents are associated with each page layout.

4 Body area layout

The body area may be subdivided into non-overlapping rectangular frames. Thus the layout may consist of any sequence of:

- a) Single frame of fixed width, equal or less than body area width, and fixed height or height adjustable to fit contents
- b) Set of multiple column frames of fixed widths per column and fixed height or height adjustable to fit contents
- c) Individual frames with fixed position and dimensions
- d) Mixed set of frames with various properties, e.g. fixed size figure frame with fixed sized caption frame beneath and adjustable height text frame beside both

Frames which have fixed position and dimensions are permitted to overlap.

See figure 1 for illustrations.

5 Header area layout

This is a rectangular area above the body area. It may be subdivided into a number of rectangular frames, for example to contain textual information and graphics such as a company logo.

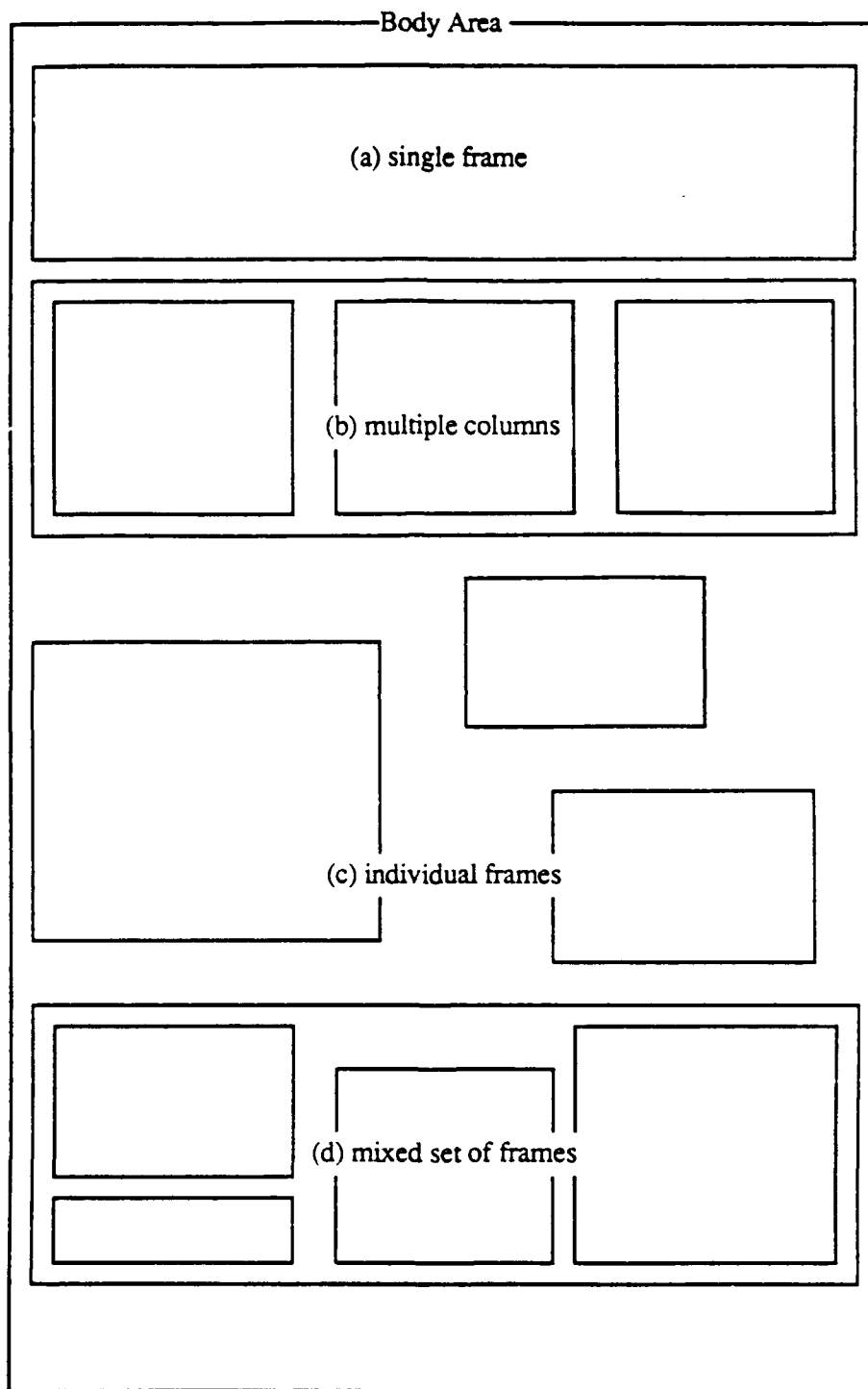


Figure 1- Examples of layout within body area

6 Footer area layout

This is a rectangular area below the body area. It may be subdivided into a number of rectangular frames, for example to contain textual information and graphics such as a company logo.

7 Header contents and footer contents

Header contents or footer contents consist of a sequence of paragraphs and/or figures that are constrained to be laid out entirely within the corresponding area.

An automatically generated page number may be included anywhere within header contents and/or footer contents.

Header contents or footer contents must not include any footnote or footnote reference.

8 Page numbering

An automatically generated page number may occur at any position within header contents or footer contents. Page numbers may be represented by Arabic numerals, lower/upper case Roman numerals or lower/upper case letters.

Page numbers are generated sequentially and the sequence can be restarted from any positive integer value at the beginning of any page set. A sub-sequence can be inserted for the purpose of amendment page numbering, e.g., ...6 7 7a 7b 8...

9 Layout of document logical contents

The sequence of passages and/or sections is laid out in one or more body areas such that it flows through the sequence of pages in the document.

Controls are needed in order to break the flow of contents at appropriate points. For example, following the passages to be placed on the title page of a document it may be required to control the flow in order to direct subsequent text onto a new page of a different page layout.

10 Layout of section contents

A section can be laid out in any of these ways:

- a) As a separate passage (see below)
- b) Below the previous text within a containing passage
- c) As a sequence of passages

11 Layout of passage contents

Controls are available to guide the layout of passages or their subordinate paragraphs and figures.

A passage can be positioned at a fixed position (e.g. the start) of a new body area or in a new frame below the previous contents of a body area.

In case of multiple columns, content generally flows from the bottom of one column of the set to the top of the next column to the right.

Regardless of content type, the various paragraphs and figures in a passage can be laid out within specified frames.

The various methods of subdivision of body areas may be combined with certain frames being designated for flowing text and other frames for particular contents. Thus text may appear to flow around other contents. For example, several figures can be contained within a passage and effect of text flow around the figures and their captions can be produced. See figure 2 for illustration.

A new set of multiple frames can occur beneath a similar set. Thus parallel text (e.g. multilingual) can be synchronized or a table effect can be generated. See figure 3 for illustration.

A variation of the table technique can be used for labelling and annotating paragraphs.

A complete passage can be constrained to be contained in the same body area or frame (by indivisibility).

12 Layout controls

The following properties may be specified to control where body area or page breaks occur:

a) New column set (New Layout Object)

This specifies that the contents should be laid out in the first column (or frame) of a new set of columns (or frames)

b) Unconditional column break (New Layout Object)

This indicates that the contents must be displayed in the next column (or frame)

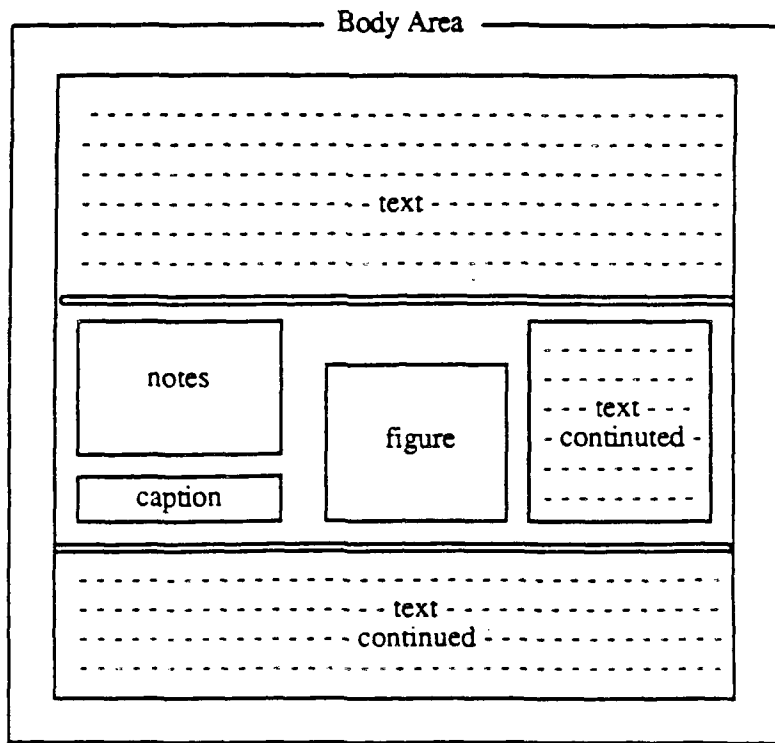


Figure 2 - Example of text flow around figure

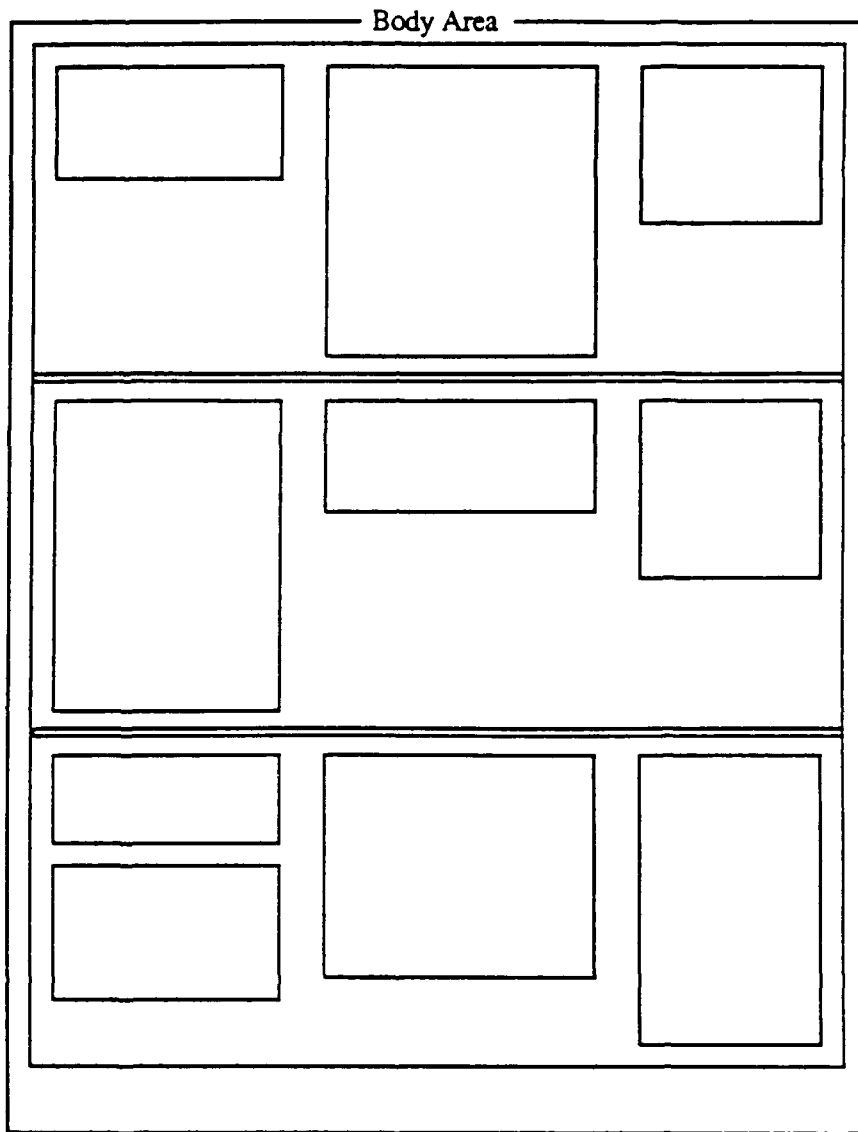


Figure 3 - Example of synchronized text.

c) Layout object class

This indicates that the contents concerned must be displayed in a specified frame, e.g. to control figure positioning

d) New page set (New Layout Object)

This indicates that the contents should be laid out in a new page set

e) New page layout (New Layout Object)

This indicates that the contents should be laid out on a new page of a particular page layout

f) Unconditional page break (New Layout Object)

This indicates that the contents must be displayed in the body area of the next page.

g) Indivisibility

This indicates that a passage (section, paragraph or figure) must be laid out within a single frame, body area or page set.

h) Same page/same area

This specifies that the start of a passage (section, paragraph or figure) must be laid out in the same frame or body area as the end of the previous content (for example to keep a first paragraph with a section title)

13 Layout of paragraph contents

A paragraph may or may not specify its own margins, alignment and tab stops. The indentation of the first line may be different from the remainder of the paragraph. The separation between successive paragraphs can be controlled.

Within a passage the contents of a paragraph can be laid out in two or more frames to give the effect of text wrapping around a figure. The figure may or may not be logically associated with that paragraph.

Layout of paragraphs can be directed by the controls described above or by the following additional control:

widow and orphan

Note: The widow and orphan feature controls where breaks may occur within the body of a paragraph.

The orphan size specifies the minimum number of lines of text that must be allocated to the first body area or frame.

The widow size specifies the minimum number of lines of text that must be allocated to the last body area or frame when a paragraph is split over two or more frames.

14 Layout of figure contents

A figure can occur beneath the previous contents of a body area or frame or can be specified to occupy a particular frame within the layout of a passage.

Any paragraphs associated with the figure in order to provide captions or notes can be positioned to occupy rectangular areas positioned above, below or beside the figure.

15 Layout of footnote contents

A footnote body is placed at the bottom of a body area of a page and is constrained to be entirely in the same body area as the reference to it. If multiple footnotes occur in the same body area the corresponding footnote bodies are placed in the body area in the same order as the reading order of their references.

12.6.1.3 Content Characteristics

The content characteristics of this Document Application Profile are:

1. Raster graphics contents, as detailed in the specification of Group 3 and Group 4 facsimile (CCITT Recommendations T4 and T6);
2. Geometric graphics contents, as detailed in the minimum capabilities defined for the Computer Graphics Metafile standard (ISO 8632);
3. Character contents, as detailed below

Character presentation

Character presentation is controlled by the presentation attributes specified in [ISO 8613-6.2]. Their basic values are summarized below for convenience of reference.

1 First line format

This produces one of the following effects:

- a) A non-indented paragraph
- b) An indented paragraph
- c) Overhang
- d) Overhang with label

2 Line layout table

This allows a set of tabulation stop positions to be defined with alignment of "star aligned", "end aligned", "centred" or "align around".

3 Character path

This normally from left to right (0) but the text of a paragraph may be specified to run from the bottom towards top of page (90).

4 Alignment

This specifies that the lines of text are to be "start-aligned", "end/aligned", "centred", or "justified".

5 Line spacing

For fonts with constant height the basic values are 3, 4, 6, 8 or 12 lines per 25.4mm.

6 Character spacing

For fonts with constant spacing the basic values are 10, 12, or 15 characters per 25.4mm.

7 Font selection

This allows selection from up to 10 fonts, including proportionally spaced fonts.

8 Graphic rendition

This allows graphic characters to be presented with a mode of emphasis selected as "normal rendition", italicized", "increased intensity (bold)", "crossed-out", "underlined", or "double underlined".

Character set features and control functions

The list of features and control functions supported includes the following.

The effects of font selection, graphic rendition, character spacing and line spacing can be changed at any point within the text of a paragraph.

Sequences of characters within a line may be subscripted or superscripted.

Text can be aligned with specific tabulation stops.

Text strings can be terminated by a required newline and can be word wrapped within the paragraph margins.

Non-breaking spaces are supported.

Discretionary hyphens are supported.

12.6.1.4 Document profile features

A document profile is associated with the document to provide information to handle it as a whole.

The features supported by this document application profile include all document management attributes defined in [ISO 8613/4].

12.6.2 NOTATION AND CONVENTIONS

12.6.2.1 Notation

The value description of attributes indicates by an asterisk (*) before the attribute value description when the value specified for an object class may be overridden by the value specified for any object of the class. In all other cases the value cannot be overridden.

When the value description specifies "-- any value" this is to be interpreted as meaning any of the values defined as permissible values by ISO 8613.

The notation used to specify attribute values is that of Annex A of ISO DIS 8613/2.2 in all cases where an appropriate notation is defined in that annex (i.e., for construction expressions, string expressions, numeric expressions, object identifier expressions, bindings, references to binding values).

12.6.2.2 Superclasses

The superclass defined in Section 12.6 specifies all the possible generic and specific logical and layout structures that can be interchanged between systems conforming to this NBS Implementor's Agreement.

The generic structures in this implementor's agreement are always complete generic structures. The specific structures must always be instances of the superclass object descriptions. That is, the values of attributes applicable to object descriptions and their associated styles must be specified within the range of permissible values defined for any corresponding object superclass description. Further, for some specified attributes of particular object superclass descriptions the values in corresponding specific objects must not override values specified by the corresponding generic object class description. External documents and resource document, if used, must conform to the superclass definition.

The superclass is defined both diagrammatically and by way of tables that list all the permissible values of attributes applicable to object class descriptions, object descriptions, and associated styles.

12.6.2.3 Superclass Expressions

Iter, ser, set, any and poss are construction operators used to define the permissible values of the construction expressions in the attribute "generator for subordinates" of all object classes of the superclass.

- iter a construction operator used to indicate that the superclass expression always evaluates to a sequence of instances of the contained superclass expression. Each instance can correspond to a different evaluation of the superclass expression.
- ser a construction operator used to indicate that the superclass expression always evaluates to a sequence of one instance of each of the contained superclass expressions. The instances occur in the sequence in the same order as the contained superclass expressions are specified.
- any a construction operator which is used to indicate that the superclass expression always evaluates to an instance of one of the contained superclass expressions.
- poss a construction operator which is used to indicate that the superclass expression optionally evaluates to either the empty sequence of a superclass expression or to an instance of the contained superclass expression.
- set a construction operator used to indicate that the superclass expression always evaluates to a sequence of one instance of each of the contained superclass expressions. The instance can occur in any order.

The following rules apply to construction operators applying to a contained superclass expression including an empty sequence.

```

any(<--->,empty sequence,<...>) = any(<--->,<...>)
ser(<--->,empty sequence,<...>) = ser(<--->,<...>)
set(<--->,empty sequence,<...>) = set(<--->,<...>)
poss(empty sequence)           = empty sequence
iter(empty sequence)           = empty sequence
any (empty sequence)           = empty sequence
ser (empty sequence)           = empty sequence
ser (empty sequence)           = empty sequence

```

- opt a construction operator used to indicate that the superclass expression always evaluates to an optional construction factor which is an instance of the contained superclass expression.

- rep a construction operator used to indicate that the superclass expression always evaluates to a repetitive construction factor which is an instance of the contained superclass expression.
- cho a construction operator used to indicate that the superclass expression always evaluates to a choice construction factor each item of which is an instance of one of the contained superclass expressions.
- seq a construction operator used to indicate that the superclass expression always evaluates to a sequence construction factor each item of which is an instance of one of the contained superclass expressions.
- agg a construction operator used to indicate that the superclass expression always evaluates to an aggregate construction factor each item of which is an instance of one of the contained superclass expressions.
- opt rep a construction operator used to indicate that the superclass expression always evaluates to a repetitive construction factor which is an instance of the contained superclass expression.

12.6.2.4 Use of Binding Expressions

This document application profile permits bindings to be used for automatic numbering schemes, e.g., page numbers and section numbers. This section describes the conventions to be used.

The superclass object specifications identify bindings by names which describe the use of each binding. Any number of bindings may be used corresponding to each name. In order to simplify recognition of bindings, their identifier values must be allocated as follows (where n is any integer) :

<u>binding name</u>	<u>identifier value</u>
number	8n+1
number string	8n+2
prefix	8n+3
suffix	8n+4
separator	8n+5

12.6.2.4.1 Use of bindings to construct sequential numbers

The binding "number string" of the numbered object is used to

construct the character string representation of the number.

If the numbered objects are all of the same object class, the ORDINAL() numeric function application can be used to create the sequence. If the numbered objects can be of different object classes, sequences are generated by incrementing the value of another binding called "number".

The "number string" binding is referenced by a content generator in a subordinate of the numbered object.

```
number          ::= INC(B_REF(PREC_OBJ(CURR_OBJ))(number))
number string   ::= <hierarchic exprn> | <simple exprn>
<hierarchic exprn> ::= B_REF(SUP_OBJ(CURR_OBJ))(number string)
                   + B_REF(SUP_OBJ(CURR_OBJ))(separator)
                   + <simple exprn>
<simple exprn>   ::= <string function>
                   (B_REF(CURR_OBJ)(number))
                   | <string function> (ORD(CURR_OBJ))
<string function> ::= MK_STR | U_ALPHA | L_ALPHA |
                   U_ROM | L_ROM
Content Generator ::= <num st> | <pre st> + <num st> |
                   <num st> + <suf st> |
                   <pre st> + <num st> + <suf st>
<num st>        ::= B_REF(SUP_OBJ(CURR_OBJ))(number string)
<pre st>        ::= B_REF(SUP_OBJ(CURR_OBJ))(prefix)
                   | <string literal>
<suf st>        ::= B_REF(SUP_OBJ(CURR_OBJ))(suffix)
                   | <string literal>
```

12.6.2.4.2 Initialization of numbering factors

A "number string" binding must be initialized in an object superior to the relevant numbering scheme (e.g., a passage can initialize a numbering scheme for subordinate sections).

```
number string   ::= " "
```

A "number" binding, if used, is initialized at each hierarchical level (e.g., section) to start the numbering sequence for subordinates.

```
number          ::= <non-negative integer>
```

The "prefix", "separator" and "suffix" bindings must be initialized at a level above the numbering scheme and can be respecified at any level within the numbering scheme.

```
prefix          ::= <string literal>
suffix          ::= <string literal>
separator       ::= <string literal>
```

12.6.2.5 Object class identifiers

In order to facilitate recognition of structures and semantics, "Object Class Identifiers" must be specified in accordance with a convention that relates them to the relevant object superclass.

With the exception of Document Logical Root and Document Layout Root, all object class identifiers must be specified as a sequence of at least three integers. The first two integers uniquely identify the superclass according to the table below. The remaining integers may be any value to uniquely identify the object class.

Passage	2	2	Pageset	0	2
Section	2	3	Page	0	3
Number	2	4	RPage	0	4
Title	2	5	VPage	0	5
Text&Refs	2	6	HDR	0	6
FNote	2	7	FTR	0	7
FNBody	2	8	BodyFR	0	8
Figure	2	9	FrameA	0	9
Text	2	10	FrameB	0	10
Graphic	2	11	FrameC	0	11
Reference	2	12	FrameD	0	12
Ref	2	13	FrameE	0	13
Raster	2	14	FrameF	0	14
Geometric	2	15	FrameG	0	15
			FrameH	0	16
			Block	0	17

Diagram of Logical Structure

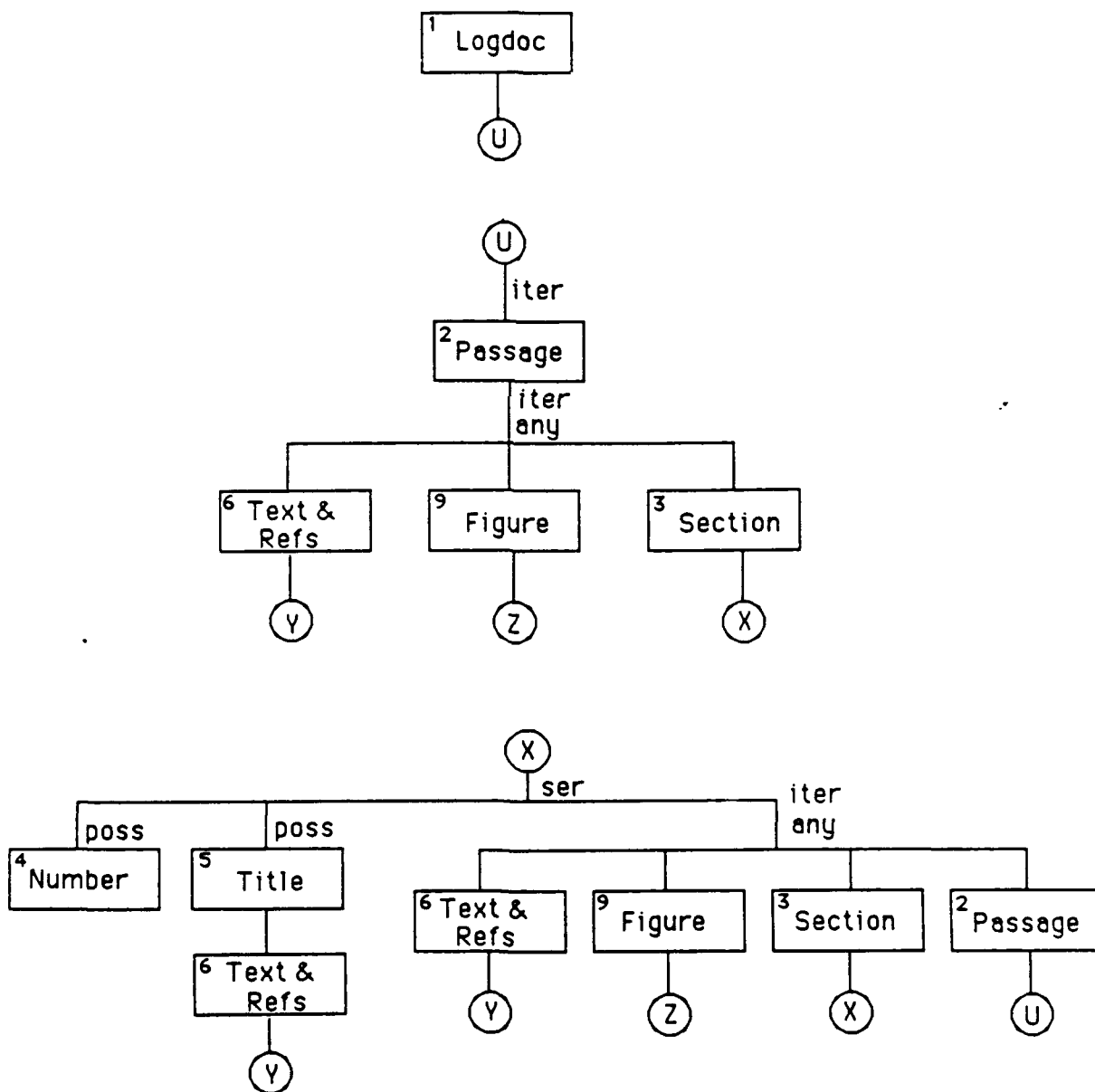
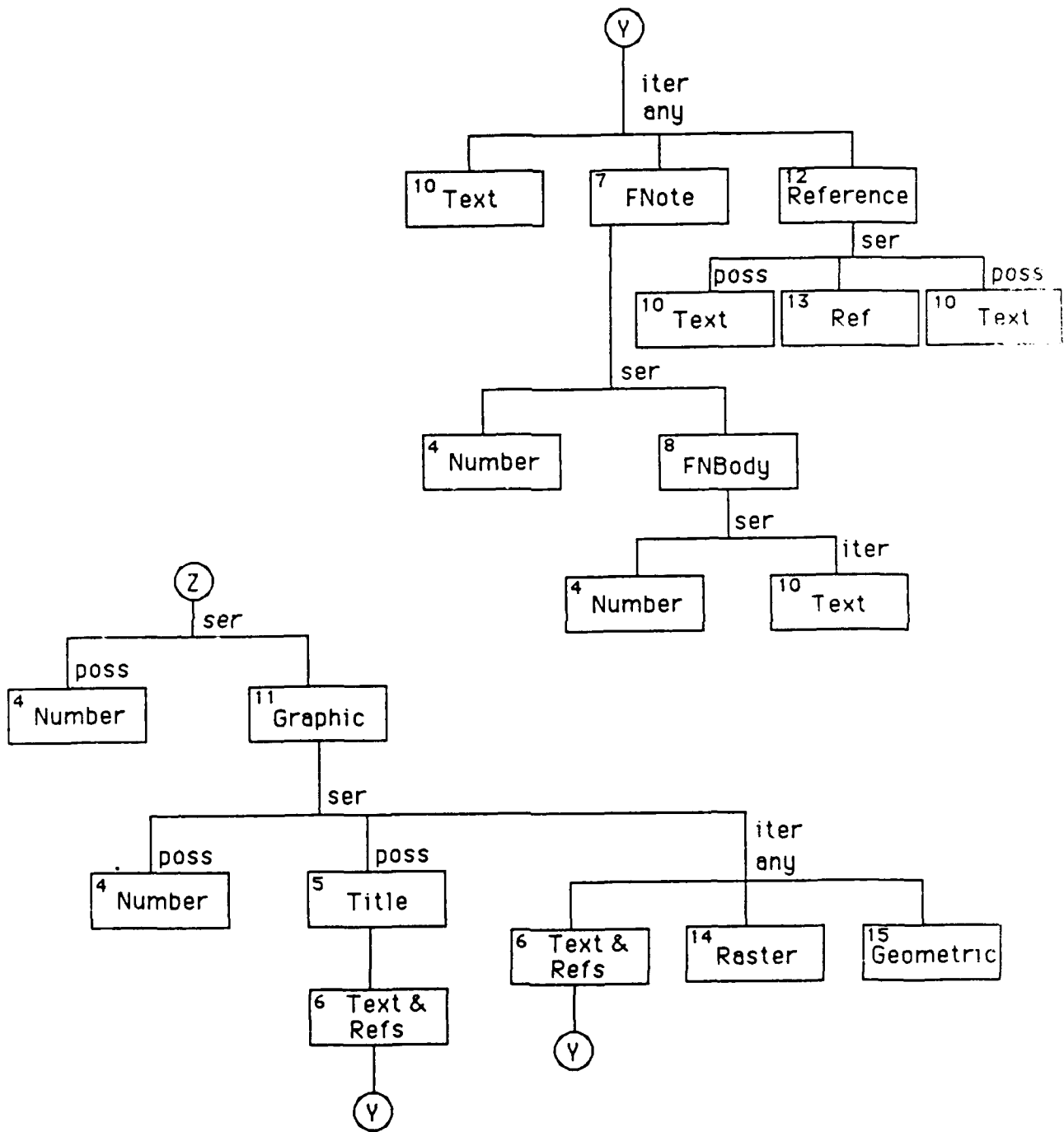


Diagram of Logical Structure (continued)



12.6.3 Logical Components

This section contains definitions of the superclass objects shown in the diagram called "Diagram of the Logical Structure."

12.6.3.1 Superclass Name: Logdoc (Logical Document)

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'document logical root'
Object Class Identifier	2
Generator for Subordinates	iter(Passage)
Layout Style	-- any value
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- initialization of any: number string, number, prefix, suffix, separator
Default Value Lists	-- any value
Protection	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value
Layout Object Class	0

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.2 Superclass Name: Passage

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- Passage
Generator for Subordinates	iter(any(Text&Refs, Figure, Section))
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- initialization of any: number string, number, prefix, suffix, separator
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.3 Superclass Name: Section

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- Section
Generator for Subordinates	ser(poss(Number),poss(Title), iter(any(Text&Refs,Figure,Section, Passage)))
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- initialization of any: separator, prefix, suffix, number (for subordinates); use of number and/or number string (to generate number)
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.4 Superclass Name: Number

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'basic logical object'
Object Class Identifier	-- Number
Content Generator	-- see Section 12.6.2
Content Portions	-- any value
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
Presentation Style	-- any value
Content Architecture Class	ASN.1 object identifier for character formatted content architecture
User Readable Comments	-- any value
User Visible Name	-- any value
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Block Alignment	-- any value
Concatenation	-- any value
Indivisibility	-- any value
Layout Category	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Offset	-- any value
Same Layout Object	-- any value
Separation	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Required Presentation Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
-----------------------	--------------------------

Presentation Style Identifier	-- any value
-------------------------------	--------------

Permitted Presentation Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
-----------------------	--------------------------

User Readable Comments	-- any value
User Visible Name	-- any value
Layout Texture	-- any value
Border	-- any value
Character Content Presentation Attributes	-- see Section 12.7.1.1 (Character Formatted)

12.6.3.5 Superclass Name: Title

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- Title
Generator for Subordinates	poss(Text,Refs,Footnote)
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Content Generator	-- see Section 12.6.2
Resource	-- any value
Presentation Style	-- any value
Content Architecture Class	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- see Section 12.6.2
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.6 Superclass Name: Text&Refs

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- Text&Refs
Generator for Subordinates	ser(iter(any(Text,FNote,Reference)))
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- initialization (for numbering subordinate items, references, figures, etc.) of any: separator, prefix, suffix, number
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.7 Superclass Name: FNote (Footnote)

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- FNote
Generator for Subordinates	ser(FNumber, FNBody)
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- use of number and/or number string

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.8 Superclass Name: FNBody (Footnote Body)

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- FNBody
Generator for Subordinates	ser(Number,Text)
Layout Style	-- any value
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- see Section 12.6.2
Protection	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Object Class	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Indivisibility	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.9 Superclass Name: Figure

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- Figure
Generator for Subordinates	ser(poss(Number),Graphic)
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- use of number and/or number string
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.10 Superclass Name: Text

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'basic logical object'
Object Class Identifier	-- Text
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
Presentation Style	-- any value
Content Architecture Class	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Protection	-- any value
Layout Style	-- any value
Content Portions	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Block Alignment	-- any value
Concatenation	-- any value
Fill Order	-- any value
Indivisibility	-- any value
Layout Category	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Offset	-- any value
Same Layout Object	-- any value
Separation	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Required Presentation Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Presentation Style Identifier	-- any value

Permitted Presentation Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
User Readable Comments	-- any value
User Visible Name	-- any value
Layout Texture	-- any value
Border	-- any value
Character Content Presentation Attributes	-- see Sections 12.7.1.2 and 12.7.1.3

12.6.3.11 Superclass Name: Graphic

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- Graphic
Generator for Subordinates	ser(poss(Number),poss(Title),iter (any(Text&Refs,Raster,Geometric)))
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- initialization (for numbering subordinate items, references, figures, etc.) of any: separator, prefix, suffix, number
Default Value Lists	-- any value
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.12 Superclass Name: Reference

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'composite logical object'
Object Class Identifier	-- Reference
Generator for Subordinates	ser(poss(Text),Ref,poss(Text))
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates of the object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Bindings	-- use of number and/or number string
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Same Layout Object	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object.

12.6.3.13 Superclass Name: Ref

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'basic logical object'
Object Class Identifier	-- Ref
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Content Generator	-- see Section 12.6.2
Content Portions	-- any value
Resource	-- any value
Presentation Style	-- any value
Content Architecture Class	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Block Alignment	-- any value
Concatenation	-- any value
Fill Order	-- any value
Indivisibility	-- any value
Layout Category	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Offset	-- any value
Same Layout Object	-- any value
Separation	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Required Presentation Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
-----------------------	--------------------------

Presentation Style Identifier	-- any value
-------------------------------	--------------

Permitted Presentation Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
-----------------------	--------------------------

User Readable Comments	-- any value
User Visible Name	-- any value
Layout Texture	-- any value
Border	-- any value
Character Content Presentation Attributes	-- see Sections 12.7.1.2 and 12.7.1.3

12.6.3.14 Superclass Name: Raster

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'basic logical object'
Object Class Identifier	-- Raster
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Content Portions	-- any value
Content Architecture class	ASN.1 object identifier for Raster Graphics Content Architecture
Resource	-- any value
Presentation Style	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Block Alignment	-- any value
Indivisibility	-- any value
Layout Category	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Offset	-- any value
Same Layout Object	-- any value
Separation	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value

Presentation Style Attributes

User Readable Comments	-- any value
User Visible Name	-- any value
Layout Texture	-- any value
Border	-- any value
Raster Graphics Content	-- see Section 12.7.2
Presentation Attributes	

12.6.3.15 Superclass Name: Geometric

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'basic logical object'
Object Class Identifier	-- Geometric
Object Identifier	-- any value
Object Class	-- identifier of object class of this superclass

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Content Portions	-- any value
Content Architecture Class	ASN.1 object identifier for Geometric Graphics Content Architecture
Resource	-- any value
Presentation Style	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Protection	-- any value
Layout Style	-- any value

Required Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Layout Style Identifier	-- any value

Permitted Layout Style Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Indivisibility	-- any value
Layout Category	-- any value
Layout Object Class	-- any value
New Layout Object	-- any value
Offset	-- any value
Same Layout Object	-- any value
Separation	-- any value
Synchronization	-- any value
User Readable Comments	-- any value
User Visible Name	-- any value
Block Alignment	-- any value

Presentation Style Attributes

User Readable Comments	-- any value
User Visible Name	-- any value
Layout Texture	-- any value
Border	-- any value
Geometric Graphics Content Presentation Attributes	-- see Section 12.7.3

Diagram of Layout Structure

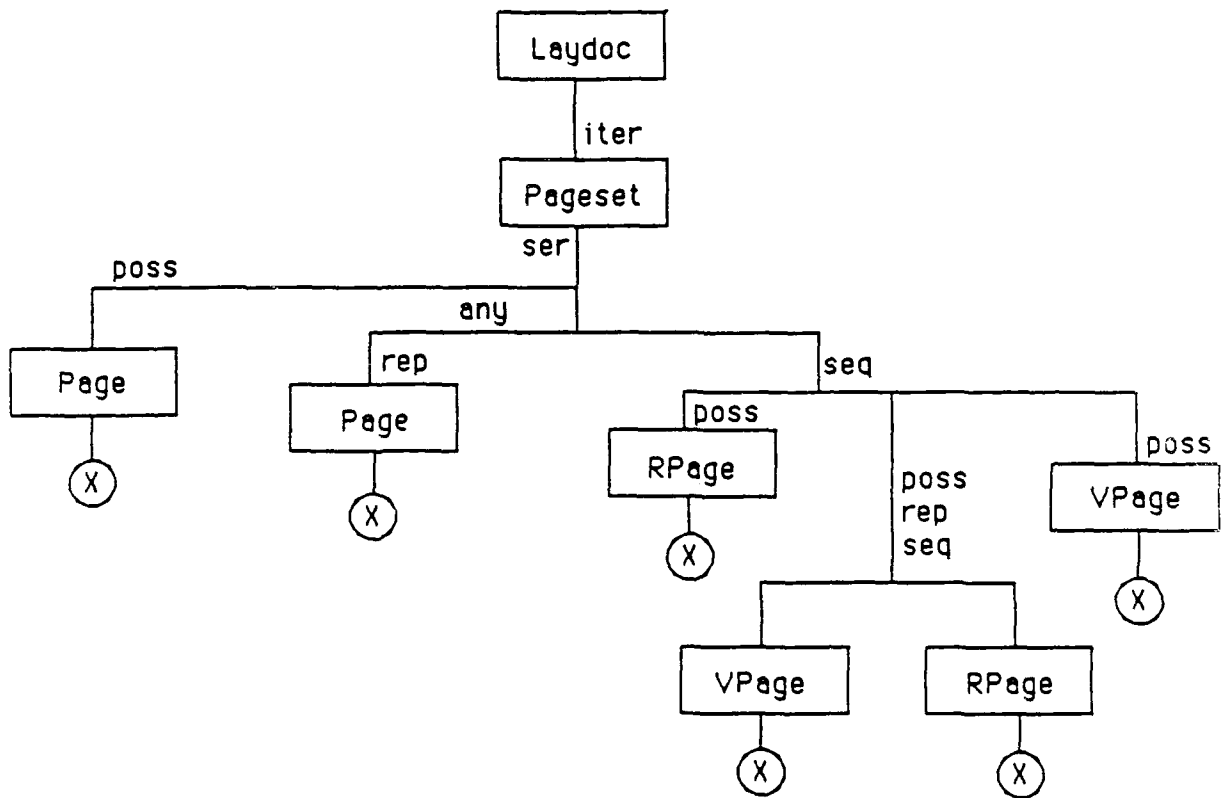
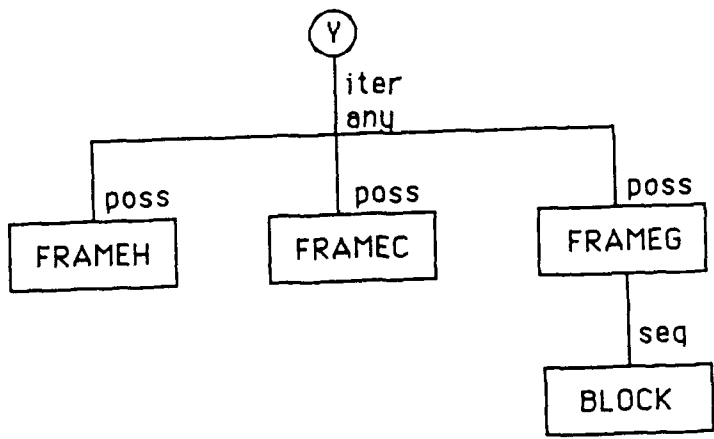
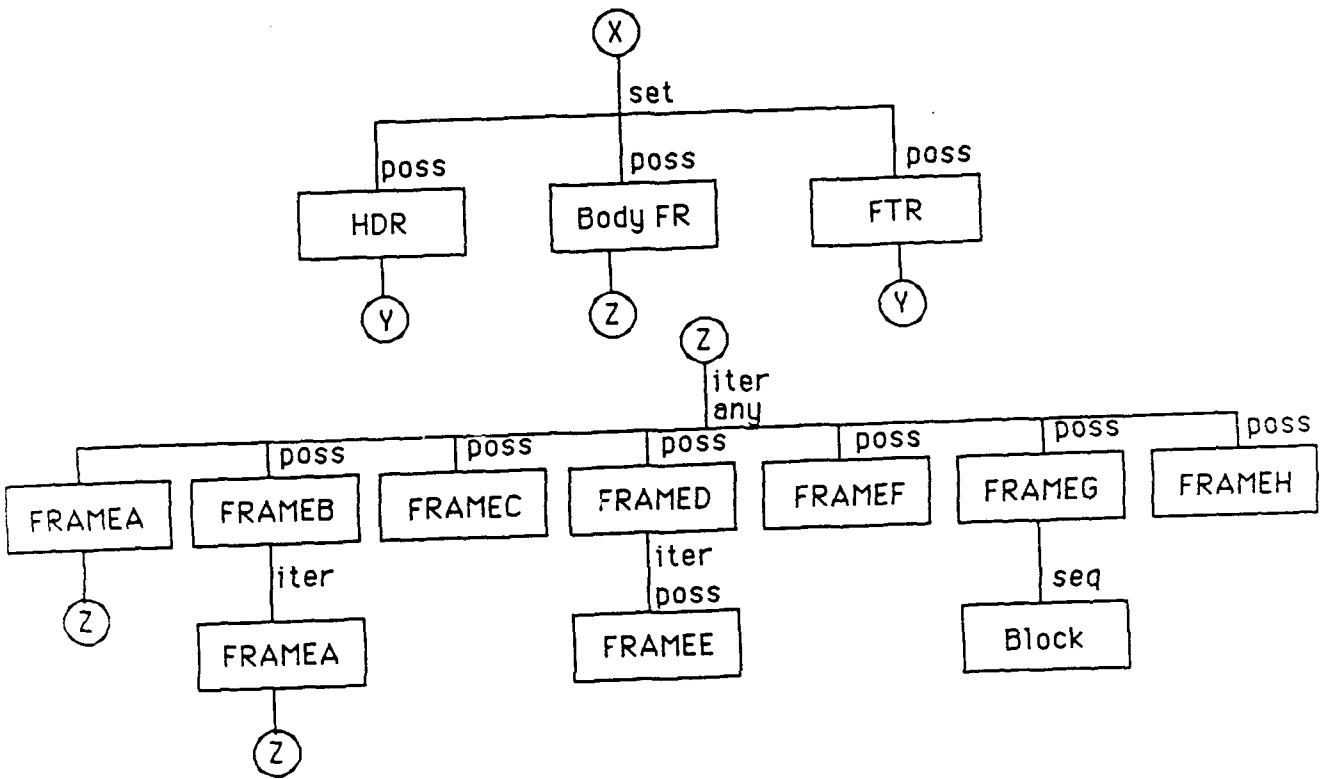


Diagram of Layout Structure (continued)



12.6.4 Layout Components

This section contains definitions of the superclass objects shown in the diagram called "Diagram of the Layout Structure."

12.6.4.0 Superclass Name: Laydoc

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	'document layout root'
Object Class Identifier	0
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: iter(PageSet)
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Default Value Lists	-- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.1 Superclass Name: Pageset

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Pageset
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: ser(poss(PAGE), any(REP(PAGE),SEQUENCE(poss(RPAGE),poss(rep(SEQUENCE(VPAGE,RPAGE))), poss(VPAGE))))
	Note: Each of the two instances of RPAGE refer to the same generic object class. Similarly, each of the two instances of VPAGE refer to the same generic object class.
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates
Bindings	PgNum,<numeric literal>

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

<u>Binding</u>	<u>Description</u>
PgNum	Initializes Page Number

12.6.4.2 Superclass Name: Page

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Page
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: set(poss(HDR), poss(BodyFR), poss(FTR))
Bindings	PgNum, Inc(B-Ref(PREC (CURR-OBJ))(PgNum))
Dimensions	-- dimensions are any of the assured reproduction areas for ISO A4, ISO A3 or North American letter.
Medium Type	
Nominal Page Size	-- any value
Side of Sheet	'unspecified'
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Layout Texture	-- any value
Page Position	-- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

Binding

Description

PgNum

Increment Page Number

12.6.4.3 Superclass Name: RPage

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Page
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: set(poss(HDR) poss(BodyFR) poss(FTR))
Bindings	PgNum, Inc(B-Ref(PREC (CURR-OBJ))(PgNum))
Dimensions	-- dimensions are any of the assured reproduction areas for ISO A4, ISO A3 or North American letter.
Medium Type	
Nominal Page Size	-- any value
Side of Sheet	'recto'
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Layout Texture	-- any value
Page Position	-- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

Binding Name Description

PgNum Increment Page Number

12.6.4.4 Superclass Name: VPage

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Page
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: set(poss(HDR) poss(BodyFR) PgNum, Inc(B-Ref(PREC (CURR-OBJ)) (PgNum)) poss(FTR))
Bindings	
Dimensions	-- dimensions are any of the assured reproduction areas for ISO A4, ISO A3 or North American letter.
Medium Type	
Nominal Page Size	-- any value
Side of Sheet	'verso'
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Layout Texture	-- any value
Page Position	-- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

Binding

Description

PgNum

Increment Page Number

12.6.4.5 Superclass Name: Header (HDR)

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (composite)
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: iter (any(poss(FrameH), poss(FrameC), poss(FrameG)))
Position	-- any constant value
Dimensions	x=<any>.y=<any> -- any constant value h=<any>,v=<any>
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	* -- any value
Layout Path	-- any value
Permitted Categories	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.6 Superclass Name: Footer (FTR)

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (composite)
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: iter(any(poss(FrameH), poss(FrameC),poss(FrameG)))
Position	-- any constant value x=<any>,y=<any>
Dimensions	-- any constant value h=<any>,v=<any>
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	-- any value
Layout Path	-- any value
Permitted Categories	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.7 Superclass Name: BodyFR

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (composite)
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: iter(any(poss(FrameA), poss(FrameB),poss(FrameC), poss(FrameD),poss(FrameF), poss(FrameG),poss(FrameH)))
Position	-- any constant value x=<any>,y=<any>
Dimensions	-- any constant value h=<any>,v=<any>
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	-- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.8 Superclass Name: FrameA

The use of this frame superclass is illustrated in Figure 1, case b and in ISO DIS 8613/2.2, Annex D, Clause D.1.6.

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (basic)
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: iter(any(poss(FrameA), poss(FrameB),poss(FrameC), poss(FrameD),poss(FrameF), poss(FrameG),poss(FrameH)))
Position	-- any constant values
Dimensions	x=<any>,y=<any> h=<any>;vrule b
	Note: see ISO DIS 8613/1.1, 5.4.1.2.
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- identifies a number of blocks within the frame
Permitted Categories	-- any value

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	* -- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.9 Superclass Name: FrameB

The use of this frame superclass is illustrated in Figure 1, case b and in ISO DIS 8613/2.2, Annex D, Clause D.1.6.

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (composite)
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: iter(FrameA)
Position	positioning rule (<any>) Note: see ISO DIS 8613/1.1, 5.4.1.1.
Dimensions	h=default; v=rule b Note: see ISO DIS 8613/1.1, 5.4.1.2.
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	-- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.10 Superclass Name: FrameC

The use of this frame superclass is illustrated in Figure 1, case a and in ISO DIS 8613/2.2, Annex D, Clause D.1.3.

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (basic)
Object Class Identifier	-- any value
Position	positioning rule (<any>) Note: see ISO DIS 8613/1.1, 5.4.1.1.
Dimensions	h=default; v=rule b Note: see ISO DIS 8613/1.1, 5.4.1.2.
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- identifies a number of blocks within the frame
Permitted Categories	-- any value

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	-- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.11. Superclass Name: Framed

The use of this frame superclass is illustrated in Figure 2, and in ISO DIS 8613/2.2, Annex D, Clause D.1.4 and D.1.5.

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (composite)
Object Class Identifier	-- any value
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: iter(poss(FrameE))
Position	positioning rule (<any>)
Note: see ISO DIS 8613/1.1, 5.4.1.1.	
Dimensions	h=default; v=rule a
Note: see ISO DIS 8613/1.1, 5.4.1.2.	
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- any value corresponding to the generator for subordinates

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	* -- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.12 Superclass Name: FrameE

The use of this frame superclass is illustrated in Figure 2, and in ISO DIS 8613/2.2, Annex D, Clause D.1.4 and D.1.5.

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (basic)
Object Class Identifier	-- any value
Position	positioning rule (<any>) Note: see ISO DIS 8613/1.1, 5.4.1.1.
Dimensions	h=rule b; v=rule b Note: see ISO DIS 8613/1.1, 5.4.1.2.
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- identifies a number of blocks within the frame
Permitted Categories	-- any value

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	* -- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.13 Superclass Name: FrameF

The use of this frame superclass is illustrated in ISO DIS 8613/2.2, Annex D, Clause D.1.7.

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (basic)
Object Class Identifier	-- any value
Position	x=<any>; y=(order=reversed,distance=<any>)
Dimensions	h=default; v=rule b
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- identifies a number of blocks within the frame
Permitted Categories	-- any value

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	* -- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.14 Superclass Name: FrameG

The use of this frame superclass is illustrated in Figure 1, case c and in ISO DIS 8613/2.2, Annex D, Clause D.1.6.

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (basic)
Object Class Identifier	-- any value
Position	-- any constant value
Dimensions	x=<any>,y=<any> -- any constant value h=<any>,v=<any>
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- identifies a number of blocks within the frame
Permitted Categories	-- any value

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Generator for Subordinates	-- any construction expression that is an instance of the following superclass expression: seq(BLOCK)
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	-- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.15 Superclass Name: FrameH

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Frame (basic)
Object Class Identifier	-- any value
Position	positioning rule (<any>) Note: see ISO DIS 8613/1.1, 5.4.1.1.
Dimensions	h=default; v=rule b Note: see ISO DIS 8613/1.1, 5.4.1.2.
Logical Source	-- any value
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Subordinates	-- identifies a number of blocks within the frame

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Bindings	-- one or more values, each of which initializes or increments a numbering system associated with the binding identifier
Layout Texture	-- any value
Border	-- any value
Layout Path	-- any value
Imaging Order	-- any value

Presentation Style Attributes

Presentation style attributes are not applicable to this object superclass.

Reserved Bindings

There are no standard bindings for this object superclass.

12.6.4.16 Superclass Name: Block

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Object Type	Block
Object Class Identifier	-- any value
Content Architecture Class	-- any permitted by Clause 12.7
Position	-- any constant value x=<any>,y=<any>
Dimensions	-- any constant value h=<any>,v=<any>
Object Identifier	-- any value
Object Class	-- identifier of an object class of this superclass
Content Portions	-- any value
Content Generator	-- any value

Note: One of the attributes "content portions" or "content generator" is required to be specified. Specification of both attributes for the same generic object is not permitted.

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource	-- any value
User Readable Comments	* -- any value
User Visible Name	* -- any value
Layout Texture	-- any value
Border	* -- any value
Presentation Style	-- any value
Presentation Attributes	-- any value

Presentation Style Attributes

Presentation style attributes applicable to this object superclass are to be consistent with the value specified for Content Architecture Class. All presentation attributes specified for the Content Architecture Class in clause 12.7 can be specified.

Reserved Bindings

Not applicable as the attribute bindings is not permitted.

12.6.5 Attributes

12.6.5.1 Attribute Applicability

This section identifies the attributes that are permitted for the logical and layout components, layout styles and presentation styles.

Shared Attributes

This section identifies the attributes that are permitted for both the logical and layout components. (See Table 2.)

Layout Attributes

This section identifies the attributes that are permitted on layout components. (See Table 3.)

Logical Attributes

This section identifies the attributes that are permitted on logical components.

Table 4 - Attributes which may be specified for constituents (continued), logical attributes

Logical Attributes Attribute Name	Document Logical Root	Composite Logical Object	Basic Logical Object	
Protection	NM/D	NM/D	NM/D	+++++
Layout Style	M/D	NM/D	NM/D	+++++

Layout Directive Attributes

This section identifies the attributes that are permitted for layout directives.

All layout directives are permitted in layout styles only.

Table 2 - Attributes which may be specified for constituents, shared attributes

Shared Attributes Attribute Name	Document Layout Root	Page Set	Page (Composite)	Frame	Block	Document Logical Root	Composite Logical Root	Basic Logical Object	Presentation Style	Layout Style
Object Type	M/D	M/D	M/D	M/D	M/D	M/D	M/D	M/D
Object Identifier	..M	..M	..M	..M	..M	..M	..M	..M
Object Class Identifier	M/..	M/..	M/..	M/..	M/..	M/..	M/..	M/..
Generator for Subordinates	M/..	M/..	M/..	M/..	M/..	M/..	M/..	M/..
Content Generator	NM/..	NM/..
Object Class	..M	..M	..M	..M	NM/..	..M	..M	..M
Subordinates	..M	..M	..M	..M	..M	..M	..M	..M
Content Portions	NM/NM	NM/NM
Resource	NM/..	NM/..	NM/..	NM/..	NM/D	NM/..	NM/..	M/..
Presentation Style	NM/D	NM/D
Content Architecture Class	NM/D	NM/D
Content Type
Presentation Attributes	NM	...
User-Readable Comments	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	FIM/D	IM/D	NM	NM
User-Visible Name	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM	NM
Bindings	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D	NM/D
Default Value Lists	NM/NM	NM/NM	NM/NM	NM/D	...	NM/N..	NM/NM

Key:

M: Mandatory; NM: Non-Mandatory; D: Defaultable; ..: Not applicable;

Table 3 - Attributes which may be specified for constituents (continued),
 layout attributes

Layout Attributes Attribute Name	Document Layout Root	Page Set	Page (Composite)	Frame	Block
Position	---	---	---	NM/D	NM/D
Dimensions	---	---	NM/D	NM/D	NM/D
Layout Texture	---	---	NM/D	NM/D	NM/D
Border	---	---	---	NM/D	NM/D
Balance	NM/D	NM/D	NM/D	NM/D	---
Layout Path	---	---	---	NM/D	---
Logical Source	---	---	---	NM/--	---
Permitted Categories	---	---	---	NM/D	---
Imaging Order	---	---	--/NM	--/NM	---
Page Position	---	---	NM/D	---	---
Medium Type	---	---	NM/D	---	---

Key:

M : Mandatory; NM : Non-Mandatory; D : Defaultable; -- : Not Applicable

Table 5 Attributes which may be specified for constituents
(continued), layout directive attributes

Layout Directives Attribute Name	Document Logical Root	Composite Logical Object	Basic Logical Object	Layout Style
- Layout Directive Attributes				
++ Block alignment	----	----	----	NM
Concatenation	----	----	----	NM
Fill order	----	----	----	NM
Indivisibility	----	----	----	NM
Layout Category	----	----	----	NM
Layout Object Class	----	----	----	NM
New layout object	----	----	----	NM
Offset	----	----	----	NM
Same layout object	----	----	----	NM
Separation	----	----	----	NM
++ Synchronization	----	----	----	NM

Key:

M : Mandatory; NM : Non-Mandatory; D : Defaultable;
-- : Not applicable

Layout Style Attributes

This section identifies the attributes that are permitted for layout styles.

Table 6 Attributes which may be specified for constituents
(continued), layout style attributes

Layout Style Attributes Attribute Name	Layout Style
Layout Style Identifier	M
User-Readable Comments	NM
User-Visible Name	NM

Presentation Style Attributes

This section identifies the attributes that are permitted for presentation styles.

Table 7 Attributes which may be specified for constituents
(continued), presentation style attributes

Presentation Style Attributes Attribute Name	Presentation Style
Presentation Style Identifier	M
Presentation Attributes	NM
User-Readable Comments	NM
User-Visible Name	NM
Layout Texture	NM
Border	NM

Content Portion Attributes

This section identifies the attributes that are permitted for content portion attributes.

Table 8 Attributes which may be specified for constituents
(concluded), content portion attributes

Content portion Attributes Attribute Name	Content portion
Content identifier - logical	M
Content identifier - layout	M
Type of Coding	NM
Content information	NM
Alternative Representation	NM
Coding attributes	*

* Classification defined in each content architecture.

Key:

M : Mandatory; NM : Non-Mandatory; D : Defaultable;
-- : Not applicable;

12.6.5.2 Attribute Values

This section defines the basic, non-basic and default values for the allowed attributes.

Shared Attributes

The attributes defined in this section can be specified for logical and/or layout components.

Attribute	Basic Value	Non-Basic Value	Default Value
Object Type	Any	None	None
Object ID	As defined in 8613/2.2	None	None
Object Class Id	As defined in 8613/2.2	None	None
Generator for Subordinates	Construction Expression	None	None +++
Content Generator	As defined in 8613/2.2	None	None
Object Class	As defined in 8613/2.2	None	None
Subordinates	As defined in 8613/2.2	None	None
Content Portions	As defined in 8613/2.2	None	None
Resource	As defined in 8613/2.2	None	None
Presentation Style	As defined in 8613/2.2	None	None
Content Architecture Class	As defined in 8613/2.2	None	None
Content Type	Not Supported In this Profile		
User-readable Comments	As defined in 8613/2.2	None	None

Attribute	Basic Value	Non-Basic Value	Default Value
User-readable Name	As defined in 8613/2.2	None	None
Bindings	As defined in 8613/2.2	None	None
Default Value Lists	As defined in 8613/2.2	None	None

Layout Attributes

The attributes defined in this section can be specified for layout components.

Attribute	Basic Value	Non-Basic Value	Default Value
Position	As defined in 8613/2.2	None	None
Dimensions	As defined in 8613/2.2	None	None

Exception: Subparameter "variable page height" of parameter "vertical dimension" is not supported.

Layout Texture	As defined in 8613/2.2	None	None
Border	As defined in 8613/2.2	None	None
Balance	As defined in 8613/2.2	None	None
Layout Path	0, 90, 180 270	None	270
Logical Source	As defined in 8613/2.2	None	None
Permitted Categories	As defined in 8613/2.2	None	None

Attribute	Basic Value	Non-Basic Value	Default Value
Imaging Order	As defined in 8613/2.2	None	None
Page Position	As defined in 8613/2.2	None	None
Medium Type	As defined in 8613/2.2	None	None

Logical Attributes

The attributes defined in this section can be specified for logical components.

Attribute	Basic Value	Non-Basic Value	Default Value
Protection	As defined in 8613/2.2	None	None
Layout Style	As defined in 8613/2.2	None	None
Layout Style ID	As defined in 8613/2.2	None	None
Block Alignment	As defined in 8613/2.2	None	None
Concatenation	As defined in 8613/2.2	None	None
Fill Order	As defined in 8613/2.2	None	None
Indivisibility	As defined in 8613/2.2	None	None
Layout Category	As defined in 8613/2.2	None	None
Layout Object Class	As defined in 8613/2.2	None	None

Attribute	Basic Value	Non-Basic Value	Default Value
New Layout Object	As defined in 8613/2.2	None	None
Offset	As defined in 8613/2.2	None	None
Same Layout Object	As defined in 8613/2.2	None	None
Separation	As defined in 8613/2.2	None	None
Synchronization	As defined in 8613/2.2	None	None

Presentation Style Attributes

The attributes defined in this section can be specified for presentation styles. Presentation styles may be applied to both logical and layout components.

Attribute	Basic Value	Non-Basic Value	Default Value
Presentation Style Id	As defined in 8613/2.2	None	None

Content Portion Attributes

The attributes defined in this section can be specified for the content portions.

Attribute	Basic Value	Non-Basic Value	Default Value
Content Id -logical -layout	As defined in 8613/2.2	None	None
Type of Coding	As defined in 8613/2.2	None	None

Attribute	Basic Value	Non-Basic Value	Default Value
Content Information	As defined in 8613/2.2	None	None
Alternative Representation	As defined in 8613/2.2	None	None

12.7 Content Architecture

This document application profile supports three content architectures: character, raster graphics, and geometric graphics. The character content architectures permit the inclusion in a document of content portions that contain graphic characters. The raster graphics content architectures permit the inclusion in a document of content portions that contain picture elements (pels). The geometric graphics content architectures permit the inclusion in a document of content portions that contain primitive graphic objects such as points, polylines, polygons, and arcs.

12.7.1 Character Content Architecture Levels

The content architecture levels defined are character formatted 3 (CF3), character processable 3 (CP3), and character formatted processable 3 (CFP3).

CF3 content architecture is an enhanced formatted form architecture which does not correspond to any existing standard and which incorporates all features defined for its class in ISO 8613.

CP3 content architecture is a processable form content architecture which does not correspond to any existing standard and which incorporates all the features defined for its class in ISO 8613.

CFP3 content architecture is a formatted processable form content architecture which does not correspond to any existing standard and which incorporates all the features defined for its class in ISO 8613.

12.7.1.1 Character Formatted

This character formatted content architecture level may be used in any basic object. Basic, non-basic and default values are specified in the following tables.

12.7.1.1.1 Presentation Attributes

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Alignment	start aligned end aligned centered justified	none	start aligned

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Alignment Indicator	not aligned aligned	none	not aligned
Character Fonts	none	any	none
Character Orientation	0 degrees	90, 180, 270 degrees	0 degrees
Character Path	0, 90 degrees	180, 270 degrees	0 degrees
Character Spacing	80, 100, 120 SMUs	any positive value	120 SMUs
First Line Format	1) indentation overhang overhang s-a item overhang e-a item 2) any non-negative value	none	indentation 0
Graphic Character Sets	The graphic character sets of ISO 6937/2 + ISO 8859/1	Any other registered character sets	ISO 6937/2
Graphic Character Subrepertoire	Subrepertoire of ISO 6937/2 equivalent to ISO 8859/1 & subrepertoire of ISO 6937/2 equivalent to Teletex	Any other registered subrepertoire of ISO 6937/2	Subrepertoire of ISO 6937/2 equivalent to ISO 8859/1
Graphic Rendition	0,1,3-4,9,10, 19,21-24,29	2, 5-7, 25-27	0
Initial Offset	1) Any non- negative value 2) Any non- negative value	none	0 120 SMU
Kerning Offset	1) Any non- negative value 2) Any non- negative value	none	0 0

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Line Layout Table	1) Any	none	no default
	2) Any	none	no default
	3) start-aligned end-aligned centered aligned-around	none	no default
	4) Any	none	no default
Line Progression	270 degrees	90 degrees	270 degrees
Line Spacing	100, 150, 200, 300, 400 SMUs	any other positive value	200 SMUs
Precision	1) Any positive value	none	1
	2) Any positive value	none	1

12.7.1.1.2 Content Elements

The graphic characters used by this content architecture level may be taken from any registered character set subject only to the restrictions defined in ISO 8613. The basic, non-basic, and default characters sets are defined by the presentation attribute "Graphic Character Sets" in 12.7.1.1.1.

12.7.1.1.3 Control Functions

Control functions with parameters

Control Functions	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Character Position Backward (HPB)	Any positive value	none	120 SMUs
Character Position Relative (HPR)	Any positive value	none	120 SMUs
Graphic Character Composition (GCC)	0, 1, 2	none	0
Identify Graphic subrepertoire (IGS)	0	Any other registered subrepertoire of ISO 6937/2	no default

Control Functions	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Line Position Backward (VPB)	Any positive value	none	100 SMUs
Line Position Relative (VPR)	Any positive value	none	100 SMUs
No Justify (JTF)	0	none	0
Select Character Spacing (SHS)	0, 1	2, 3	0
Select Graphic Rendition (SGR)	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27	0
Select Line Spacing (SVS)	0, 1, 2, 3, 4	9	0
Selective Tabulation (STAB)	Any	none	no default
Set Additional Character Spacing (SACS)	Any	none	0
Set Space Width (SSW)	Any positive value	none	if variable specify character "space", else 120 SMUs
Spacing Increment (SPI)	Any positive value Any positive value	none none	current line spacing current character spacing
Start Reverse String (SRS)	0, 1	none	0

Control functions without parameters

Backspace (BS)
 Carriage Return (CR)
 Line Feed (LF)

Partial Line Down (PLD)
 Partial Line Up (PLU)
 Space (SP)
 Substitute (SUB)

Code extension control functions

Any code extension control function defined in ISO 2022 is permitted. Interpretation and rendition of code extensions are implementation dependent.

12.7.1.1.4 Type of Coding

The value of this attribute is an ASN.1 object identifier as defined in ISO 8613 Part 6.

12.7.1.1.5 Coding Attributes

No coding attributes are defined for this content architecture level.

12.7.1.2 Character Processable

This character processable content architecture level may be used in any basic logical object. Basic, non-basic and default values are specified in the following tables.

12.7.1.2.1 Presentation Attributes

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Alignment	start aligned end aligned centered justified	none	start aligned
Character Fonts	none	any	no default
Character Orientation	0 degrees	90, 180, 270 degrees	0 degrees
Character Path	0, 90 degrees	180, 270 degrees	0 degrees
Character Spacing	80, 100, 120 SMUs	Any other positive value	120 SMUs

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
First Line Format	1) indentation overhang overhang s-a item overhang e-a item 2) any non-negative value	none	indentation 0
Graphic Character Sets	The graphic character sets of ISO 6937/2 & ISO 8859/1	Any other registered character sets	ISO 6937/2
Graphic Character Subrepertoire	Subrepertoire of ISO 6937/2 equivalent to ISO 8859/1 & subrepertoire of ISO 6937/2 equivalent to Teletex	Any other registered subrepertoire of ISO 6937/2	Subrepertoire of ISO 6937/2 equivalent to ISO 8859/1
Graphic Rendition	0, 1, 3-4, 9, 10 -19, 21-24, 29	2, 5-7, 25-27	0
Indentation	Any non-negative value	none	0
Kerning Offset	1) Any non-negative value 2) Any non-negative value	none	0 0
Leading	yes, no	none	no
Line Layout Table	1) Any 2) Any 3) start-aligned end-aligned centered aligned-around 4) Any	none none none	no default no default no default no default
Line Progression	270 degrees	90 degrees	270 degrees
Line Spacing	100, 150, 200, 300, 400 SMUs	any other positive value	200 SMUs

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Orphan Size	Any positive value	none	1
Pairwise Kerning	yes, no	none	no
Precision	1) Any positive value	none	1
	2) Any positive value	none	1
Widow Size	Any positive value	none	1

12.7.1.2.2 Content Elements

The graphic characters used by this content architecture level may be taken from any registered character set subject only to the restrictions defined in ISO 8613. The basic, non-basic, and default characters sets are defined by the presentation attribute "Graphic Character Sets" in 12.7.1.2.1.

12.7.1.2.3 Control Functions

Control functions with parameters

Control Functions	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Graphic Character Composition (GCC)	0, 1, 2	none	0
Identify Graphic Subrepertoire (IGS)	0	Any other registered repertoire of ISO 6937/2	no default
Line Position Backward (VPB)	Any positive value	none	100 SMUs
Line Position Relative (VPR)	Any positive value	none	100 SMUs
Parallel Texts (PTX)	0, 1, 3	none	0

Control Functions	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Select Character Spacing (SHS)	0, 1	2, 3	0
Select Graphic Rendition (SGR)	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27	0
Select Line Spacing (SVS)	0, 1, 2, 3, 4	9	0
Selective Tabulation (STAB)	Any	none	no default
Spacing Increment (SPI)	Any positive value Any positive value	none none	current line spacing current line spacing
Start Reverse String (SRS)	0, 1	none	0

Control functions without parameters

Break Permitted Here (BPH)
 Carriage Return (CR)
 Line Feed (LF)
 No Break Here (NBH)
 Partical Line Down (PLD)
 Partical Line Up (PLU)
 Space (SP)
 Substitute (SUB)

Code extension control functions

Any code extension control function defined in ISO 2022 is permitted. Interpretation and rendition of code extensions are implementation dependent.

12.7.1.2.4 Type of Coding

The value of this attribute is an ASN.1 object identifier as defined in ISO 8613 Part 6.

12.7.1.2.5 Coding Attributes

No coding attributes are defined for this content architecture level.

12.7.1.3 Character Formatted Processable

This character formatted processable content architecture level may be used in any basic object. Basic, non-basic and default values are specified in the following tables.

12.7.1.3.1 Presentation Attributes

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Alignment	start aligned end aligned centered justified	none	start aligned
Alignment Indicator	not aligned aligned	none	not aligned
Character Fonts	none	any	none
Character Orientation	0 degrees	90, 180, 270 degrees	0
Character Path	0, 90 degrees	180, 270 degrees	0
Character Spacing	100, 120 SMUs	Any other positive value	120 SMUs
First Line Format	1) indentation overhang overhang s-a item overhang e-a item 2) any non-negative value	none	indentation 0
Graphic Character Sets	The graphic character sets of ISO 6937/2 + ISO 8859/1	Any other registered character sets	ISO 6937/2

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Graphic Character Subrepertoire	Subrepertoire of ISO 6937/2 equivalent to ISO 8859/2 & suprepertoire of ISO 6937/2 equivalent to Teletex	Any other registered subrepertoire of ISO 6937/2	Subrepertoire of 6937/2 equivalent to ISO 8859/1
Graphic Rendition	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27	0
Initial Offset	1) Any non-negative value 2) Any non-negative value	none	0 120 SMUs
Indentation	Any non-negative value	none	0
Kerning Offset	1) Any non-negative value 2) Any non-negative value	none	0 0
Leading	yes, no	none	no
Line Layout Table	1) Any 2) Any 3) start-aligned end-aligned centered aligned-around 4) Any	none none none	no default no default no default
Line Progression	270 degrees	90 degrees	270 degrees
Line Spacing	100, 150, 200, 300, 400 SMUs	any other positive value	200 SMUs
Orphan Size	Any positive value	none	1
Pairwise Kerning	yes, no	none	no

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Precision	1) Any positive value	none	1
	2) Any positive value	none	1
Widow Size	Any positive value	none	1

12.7.1.3.2 Content Elements

The graphic characters used by this content architecture level may be taken from any registered character set subject only to the restrictions defined in ISO 8613. The basic, non-basic, and default characters sets are defined by the presentation attribute "Graphic Character Sets" in 12.7.1.3.1.

12.7.1.3.3 Control Functions

Control functions with parameters

Control Functions	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Character Position Backward (HPB)	Any positive value	none	120 SMUs
Character Position Relative (HPR)	Any positive value	none	120 SMUs
Graphic Character Composition (GCC)	0, 1, 2	none	0
Identify Graphic Subrepertoire (IGS)	0	Any other registered subrepertoire of ISO 6937/2	no default
Line Position Backward (VPB)	Any positive value	none	100 SMUs
Line Position Relative (VPR)	Any positive value	none	100 SMUs
No Justify (JTF)	0	none	0

Control Functions	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Parallel Texts (PTX)	0, 1, 3	none	0
Select Character Spacing (SHS)	0, 1	2, 3	0
Select Graphic Rendition (SGR)	0, 1, 3-4, 9, 10-19, 21-24, 29	2, 5-7, 25-27	0
Select Line Spacing (SVS)	0, 1, 2, 3,	4, 9	0
Selective Tabulation (STAB)	Any	none	no default
Set Space Width (SSW)	Any positive value	none	if variable spacing character "space", else 120 SMUs
Spacing Increment (SPI)	Any positive value Any positive value	none none	current line spacing current line spacing
Start Reverse String (SRS)	0, 1	none	0

Control functions without parameters

Backspace (BS)
 Break Permitted Here (BPH)
 Carriage Return (CR)
 Line Feed (LF)
 No Break Here (NBH)
 Partial Line Down (PLD)
 Partial Line Up (PLU)
 Space (SP)
 Start of String (SOS)
 String Terminator (ST)
 Substitute (SUB)

Code extension control functions

Any code extension control function defined in ISO 2022 is permitted. Interpretation and rendition of code extensions are implementation dependent.

12.7.1.3.4 Type of Coding

The value of this attribute is an ASN.1 object identifier as defined in ISO 8613/6.

12.7.1.3.5 Coding Attributes

No coding attributes are defined for this content architecture level.

12.7.2 Raster Graphics Content Architecture

This section specifies the assignments of attribute values for the processable content architecture class. The values are listed in tabular form. There are basic, non-basic and default values listed for each attribute.

Content architecture level RP-1 is defined for the processable form content architecture class. Content pertaining to this level may be laid out using either the fixed dimension layout method or the scalable dimension layout method of the processable content layout process.

12.7.2.1 Raster Graphics Processable

RP-1 is a content architecture level derived from the processable form content architecture class. It is laid out using either the fixed or scalable dimension methods of the processable content layout process (depending upon the value of "pel spacing"). RP-1 content form may be associated with basic layout or logical objects.

12.7.2.1.1 Presentation Attributes

Attribute	Basic Value	Non-Basic Values	Default Values
Pel Path	0, 90, 180 270 deg	None	0 deg
Line Progression	90, 270 deg	None	270 deg
Pel spacing	(Any positive integer, Any positive integer) SMU, 'null'	None	(6,1)SMU
Spacing Ratio	(Any positive integer, Any positive integer)	None	(1,1)
Clipping		None	See note 1
First Pair	(Any non-negative integer, any non-negative integer)		
Second Pair	(Any non-negative integer, any non-negative integer)		
Image Dimensions	See note 2	None	Automatic set

Attribute	Basic Value	Non-Basic Values	Default Values
Image Width Set			
Minimum Width	Any non-negative integer	None	
Preferred Width	Any non-negative integer		
Image Height Set			
Minimum Height	Any non-negative integer	None	
Preferred Height	Any non-negative integer		
Image Size Set			
Minimum Height	Any non-negative integer	None	
Preferred Height	Any non-negative integer		
Minimum Width	Any non-negative integer	None	
Preferred Width	Any non-negative integer		
Aspect Ratio	Variable, fixed	None	
Automatic Set	-	-	

Note 1: The default value of 'clipping' is the first coordinate in the content portion (0,0) and the last coordinate (N-1, L-1), where N is the number of pels per line and L is the number of lines.

Note 2: Minimum values must not be greater than the preferred values.

12.7.2.1.2 Content Elements

Content is represented in a two-dimensional pictorial image in the form of a two-dimensional array of picture elements (pels). Each element of the array comprises data used to determine the image of the corresponding pel. Each element of the array is represented by data specifying one of two states. These two states are named "set" (1) and "unset" (0). The set state is used to represent the foreground color, with the unset state used to represent the background color.

12.7.2.1.3 Control Elements

No control elements are defined within the processable raster graphics content architecture.

12.7.2.1.4 Type of Coding

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Type of Coding	Bitmap encoding T6 encoding T4 encoding	none	Bitmap encoding

12.7.2.1.5 Coding Attributes

Attribute	Basic Value(s)	Non-Basic Value(s)	Default Value(s)
Number of Pels per Line	Any positive integer	None	No default
Number of lines	Any non-negative integer	None	No default
Pel Array Order	Up, Down see note 1	None	Down

Note 1: The attribute 'pel array order' is only relevant if the attribute 'type of encoding' takes the value of 'bitmap encoding'.

12.7.3 Geometric Graphics Content Architecture

The geometric graphics content architecture permits the inclusion in documents of content portions containing graphics primitives such as lines, markers, filled areas, graphic text and etc. The content architecture is based on ISO 8632, Computer Graphics Metafile (CGM).

12.7.3.1 Geometric Graphics Formatted Processable

This section specifies the assignments of attribute values for the geometric graphics processable content architecture class, GG-0. The values are listed in a tabular form. There are basic, non-basic, and default values listed for each attribute.

12.7.3.1.1 Presentation Attributes

Geometric Graphics Encoding Announcer

Attribute	Basic Values	Non-Basic Values	Default Values
VDC Type	0 (Integer) 1 (Real)	None	0
Integer Precision	16	8,24,32	16
Real Precision	0,9,23 (Floating Point) 1,16,16 (Fixed Point)	0,12,52 (Floating Point) 1,32,32 (Fixed Point)	0,9,23 Note 1
Index Precision	16	8,24,32	16
Color Precision	8,16	24,32	8
Color Index Precision	8,16	24,32	8
Maximum Color Index	255	All other permissible values	255
Color Value Extent	Any permissible values	None	(0,255)
Color Selection Mode	0 (Indirect) 1 (Direct)	None	0

Attribute	Basic Values	Non-Basic Values	Default Values
VDC Integer Precision	16	24,32	16
VDC Real Precision	0,9,23 1,16,16	0,12,52 1,32,32	0,9,23 Note 1

Note 1: For "Real Precision", the first parameter designates the type of real. "0" implies fixed point. "1" implies floating point. The second parameter designates the bit precision of the exponent. The third parameter designates the bit precision of the fraction.

Line Rendition

Attribute	Basic Values	Non-Basic Values	Default Values
Line Width Specification Mode	0 (Absolute) 1 (Scaled)	None	1
Line Bundle Index	1-5	All other permissible values	1
Line Type	1 (Solid) 2 (Dash) 3 (Dot) 4 (Dash-dot) 5 (Dash-dot-dot)	All other permissible values	1
Line Width (If scaled)	Any permissible value	None	1.0
Line Width (If absolute)	Any permissible value	None	0.001* length of longest dimension of VDC Extent
Line Color (If indexed)	Any permissible value	None	1
Line Color (If direct)	Any permissible	None	(255,255,255)

Attribute	Basic Values	Non-Basic Values	Default Values
	value		
Line Aspect Source Flags			
Line Type	0 (Individual) 1 (Bundled)	None	0
Line Width	0 (Individual) 1 (Bundled)	None	0
Line Color	0 (Individual) 1 (Bundled)	None	0
Line Bundle Specifications			
Line Bundle Index	1-5	All other permissible values	1
Line Bundle Representation			
Line Type	Note 1	All other permissible values	Note 1
Line Width	Note 1	All other permissible values	Note 1
Line Color	Note 1	All other permissible values	Note 1

Note 1: Values for Line Bundle Representation

	Bundle Index				
	1	2	3	4	5
Line Type	1 (Solid)	2 (Dash)	3 (Dot)	4 (Dash-dot)	5 (Dash-dot-dot)
Line Width (If scaled)	1.0	1.0	1.0	1.0	1.0
(If absolute)	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent
Line Color (If indexed)	1	1	1	1	1
(If direct)	(255,255, 255)	(255,255, 255)	(255,255, 255)	(255,255, 255)	(255,255, 255)

Marker Rendition

Attribute	Basic Values	Non-Basic Values	Default Values
Marker Size	0 (Absolute)	None	1
Specification Mode	1 (Scaled)		
Marker Bundle Index	1-5	All other permissible values	1
Marker Type	1 (Dot) 2 (Plus) 3 (Asterick) 4 (Circle) 5 (Cross)	All other permissible values	3
Marker Size (If scaled)	Any permissible value	None	1.0
(If absolute)	Any permissible value	None	0.001* length of longest dimension of VDC Extent
Marker Color (If indexed)	Any permissible value	None	1
(If direct)	Any permissible value	None	(255,255,255)
Marker Aspect Source Flags			
Marker Type	0 (Individual) 1 (Bundled)	None	0
Marker Size	0 (Individual) 1 (Bundled)	None	0
Marker Color	0 (Individual) 1 (Bundled)	None	0
Marker Bundle Specifications			
Marker Bundle Index	1-5	All other permissible values	1
Marker Bundle Representation			
Marker Type	Note 1	All other	Note 1

Attribute	Basic Values	Non-Basic Values	Default Values
Marker Size	Note 1	permissible values All other permissible values	Note 1
Marker Color	Note 1	All other permissible values	Note 1

Note 1: Values for Marker Bundle Representation

	Bundle Index				
	1	2	3	4	5
Marker Type	1 (Dot)	2 (Plus)	3 (Asterick)	4 (Circle)	5 (Cross)
Marker Size (If scaled)	1.0	1.0	1.0	1.0	1.0
(If absolute)	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent
Marker Color (If indexed)	1	1	1	1	1
(If direct)	(255,255, 255)	(255,255, 255)	(255,255, 255)	(255,255, 255)	(255,255, 255)

Text Rendition

Attribute	Basic Values	Non-Basic Values	Default Values
Font List	Note 1	All other permissible values	No default
Character Set List			
Character Set	0 (94 character)	All other permissible values	0
Type	1 (96 character)		
Designation	Note 2	All other permissible	Note 2
Sequence Tail			

Attribute	Basic Values	Non-Basic Values	Default Values
		values	
Character Coding Announcer	1 (Basic 8-bit)	All other permissible values	1
Text Bundle Index	1-2	All other permissible values	1
Text Font Index	1-4	All other permissible values	1
Text Precision	0 (String) 1 (Character) 2 (Stroke)	None	2
Character Expansion Factor	Any permissible values	None	1.0
Character Spacing	Any permissible values	None	0.0
Text Color (If indexed)	Any permissible values	None	1
(If direct)	Any permissible values	None	(255,255,255)
Character Height	Any permissible values	None	0.01 * length of the longest side of the VDC Extent
Character Orientation	Note 3	All other permissible values	(0,1), (1,0)
Text Path	0 (Right) 1 (Left) 2 (Up) 3 (Down)	None	0

Attribute	Basic Values	Non-Basic Values	Default Values
Text Alignment			
Horizontal	Note 4	4	0
Vertical	Note 5	6	0
Continuous horizontal	None	All other permissible values	No default
Continuous vertical	None	All other permissible values	No default
Character Set Index			
Character Set Index	1-2	All other permissible values	1
Alternate Character Set Index			
Alternate Character Set Index	1-2	All other permissible values	1
Text Aspect Source Flags			
Text Font	0 (Individual) 1 (Bundled)	None	0
Text Precision	0 (Individual) 1 (Bundled)	None	0
Character Expansion Factor	0 (Individual) 1 (Bundled)	None	0
Character Spacing	0 (Individual) 1 (Bundled)	None	0
Text Color	0 (Individual) 1 (Bundled)	None	0
Text Bundle Specifications			
Text Bundle Index	1-2	All other permissible values	1
Text Bundle Representation			
Text Bundle Index	Note 6	All other permissible values	Note 6
Text Precision	Note 6	All other permissible values	Note 6
Character Expansion Factor	Note 6	All other permissible values	Note 6
Character Spacing	Note 6	All other permissible values	Note 6

Attribute	Basic Values	Non-Basic Values	Default Values
Text Color	Note 6	All other permissible values	Note 6

Note 1: List containing 1-4 fonts capable of representing the Subrepertoire of ISO 6937/2 equivalent to ISO 8859/1.

Note 2: Designation Sequence Tails that are registered for the character sets ISO 6937/2 and ISO 8859/1.

Note 3: Any pair of VDC vectors which have non-zero length, are not colinear and are parallel to the axis of the VDC Space.

Note 4: For Horizontal Alignment, a "0" implies "Normal Horizontal"; a "1" implies "Left"; a "2" implies "Center"; a "3" implies "Right"; a "4" implies "Continuous Horizontal".

Note 5: For Vertical Alignment, a "0" implies "Normal Vertical"; a "1" implies "Top"; a "2" implies "Cap"; a "3" implies "Half"; a "4" implies "Base"; a "5" implies "Bottom"; a "6" implies "Continuous Vertical".

Note 6: Values for the Text Bundle Representation:

	Bundle Index	
	1	2
Font Index	1	2
Text Precision	Stroke	Stroke
Character Expansion Factor	1.0	0.5
Character Spacing	0.0	0.0
Text Color (If indexed)	1	1
(If direct)	(255,255,255)	(255,255,255)

Filled Area Rendition

Attribute	Basic Value	Non-Basic Values	Default Values
Fill Bundle Index	1-5	All other permissible values	1

Attribute	Basic Value	Non-Basic Values	Default Values
Interior Style	1 (Hollow) 2 (Solid) 3 (Pattern) 4 (Hatch) 5 (Empty)	All other permissible values	1
Fill Color (If indexed)	Any permissible values	None	1
(If direct)	Any permissible values	None	(255,255,255)
Hatch Index	1-6	All other permissible values	1
Pattern Index	1-8	All other permissible values	1
Fill Reference	Any permissible values	None	(0,0)
Pattern Size			
Height Vector			
X component	Any permissible values	None	0
Y component	Any permissible values	None	Height of default VDC extent
Width Vector			
X component	Any permissible values	None	Width of default VDC extent
Y component	Any permissible values	None	Y Y
Pattern Table Specifications			
Pattern Table Index	1-8	All other permissible values	1-8
Nx	1-16	All other permissible values	1
Ny	1-16	All other permissible values	1
Local color	0,1,8,16	All other	0

Attribute	Basic Value	Non-Basic Values	Default Values
precision		permissible values	
Color (If indexed)	Any permissible values	None	1
(If direct)	Any permissible values	None	(255,255,255)
Fill Aspect Source Flags			
Interior Style	0 (Individual) 1 (Bundled)	None	0
Fill Color	0 (Individual) 1 (Bundled)	None	0
Hatch Index	0 (Individual) 1 (Bundled)	None	0
Pattern Index	0 (Individual) 1 (Bundled)	None	0
Fill Bundle Specifications			
Fill Bundle Index	1-5	All other permissible values	1
Fill Bundle Representation			
Interior Style	Note 1	All other permissible values	Note 1
Fill Color	Note 1	All other permissible values	Note 1
Hatch Index	Note 1	All other permissible values	Note 1
Pattern Index	Note 1	All other permissible values	Note 1

Note 1: Values for Fill Bundle Representation

	Bundle Index				
	1	2	3	4	5
Interior Style	4	4	4	4	4
(If indexed)	(Hatch)	(Hatch)	(Hatch)	(Hatch)	(Hatch)
Fill Color	1	1	1	1	1
(If direct)	(255,255,	(255,255,	(255,255,	(255,255,	(255,255,

	255)	255)	255)	255)	255)
Hatch Index	1	2	3	4	5
Pattern Index	1	1	1	1	1

Edge Rendition

Attribute	Basic Values	Non-Basic Values	Default Values
Edge Width Specification Mode	0 (Absolute) 1 (Scaled)	None	1
Edge Bundle Index	1-5	All other permissible values	1
Edge Type	1 (Solid) 2 (Dash) 3 (Dot) 4 (Dash-dot) 5 (Dash-dot-dot)	All other permissible values	1
Edge Width (If scaled)	Any permissible value	None	1.0
(If absolute)	Any permissible value	None	0.001* length of longest dimension of VDC Extent
Edge Color (If indexed)	Any permissible value	None	1
(If direct)	Any permissible value	None	(255,255,255)
Edge Aspect Source Flags			
Edge Type	0 (Individual) 1 (Bundled)	None	0
Edge Width	0 (Individual) 1 (Bundled)	None	0
Edge Color	0 (Individual) 1 (Bundled)	None	0
Edge Bundle Specifications			
Edge Bundle	1-5	All other	1

Attribute	Basic Values	Non-Basic Values	Default Values
Index		permissible values	
Edge Bundle Representation			
Edge Type	Note 1	All other permissible values	Note 1
Edge Width	Note 1	All other permissible values	Note 1
Edge Color	Note 1	All other permissible values	Note 1

Note 1: Values for Edge Bundle Representation

	Bundle Index				
	1	2	3	4	5
Edge Type	1 (Solid)	2 (Dash)	3 (Dot)	4 (Dash-dot)	5 (Dash-dot-dot)
Edge Width (If scaled)	1.0	1.0	1.0	1.0	1.0
(If absolute)	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent	0.001 x length of largest dimension of VDC Extent
Edge Color (If indexed)	1	1	1	1	1
(If direct)	(255,255, 255)	(255,255, 255)	(255,255, 255)	(255,255, 255)	(255,255, 255)

Color Representation

Attribute	Basic Value	Non-Basic Values	Default Values
Background Color	Any permissible values	None	(0,0,0)
Color Table Specifications			
Starting	Any permissible	None	2

Attribute	Basic Values	Non-Basic Values	Default Values
Index	values		
Color List	Any permissible values	None	Note 1

Note: The default Color Table indices 0 and 1 are explicitly defined in ISO 8632 as corresponding to the nominal background and nominal foreground colors, respectively. The following eight (8) direct color values are repeated to fill the remaining 254 entries of the color list. (1,0,0), Red; (0,1,0), Green; (0,0,1), Blue; (1,1,0), Yellow; (1,0,1), Magenta; (0,1,1), Cyan; (0,0,0), Black; (1,1,1), White.

Transparency Specification

Attribute	Basic Value	Non-Basic Values	Default Values
Transparency	0 (On) 1 (Off)	None	0
Auxiliary Color (If indexed)	Any permissible values	None	0
(If direct)	Any permissible values	None	(0,0,0)

Transformation Specification

Attribute	Basic Value	Non-Basic Values	Default Values
VDC Extent	Note 1	None	(32767, 32767)
Clip Indicator	0 (On) 1 (Off)	None	0
Clip Rectangle	Note 1	None	(32767, 32767)

Note 1: Any pair of Virtual Device Coordinates defining a rectangle.

Other Presentation Attributes

Attribute	Basic Value	Non-Basic Values	Default Values
Region of Interest	Automatic, Rectangle	None	Automatic
Picture Orientation	0,90,180,270	None	0

12.7.3.1.2 Content Elements

The value of the content portion attribute "content information" of a content portion description that conforms to ISO 8613-8 is an ASN.1 octet string representing a CGM metafile conforming to the rules defined in ISO 8632-1 with the encoding defined in ISO 8632-3.

12.7.3.1.3 Control Functions

No other control functions are defined for content portions conforming to ISO 8613-8 other than those control functions defined in ISO 8632-1 and ISO 8632-3.

12.7.3.1.4 Type of Coding

This attribute is not applicable to this document application profile.

12.7.3.1.5 Coding Attributes

No other coding attributes are defined for content portions conforming to ISO 8613-8.
values

12.8 Document Profile

Attributes that are not required or permitted cannot be used. In the tables, "any value" means any value specified in Part 4 of DIS 8613.

12.8.1 Presence of Document Constituents

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Resource Document	-- any value
Resources	-- any value
Generic Layout Structure	-- any value
Specific Layout Structure	-- any value
Generic Logical Structure	-- any value
Specific Logical Structure	-- any value
Layout Styles	-- any value
Presentation Styles	-- any value
External Document Class	-- any value

12.8.2 Document Characteristics

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Document Application Profile	-- object identifier to be supplied
Document Application Profile Defaults	-- any value
Document Architecture Class	-- any value
Content Architecture Class	-- any value
ODA Version	-- any value

12.8.2.2 Non-basic Document Characteristics

The following attributes are required when non-basic values are associated with attributes of the document.

<u>Attribute Name</u>	<u>Value Description</u>
Profile Character Sets	Note 1
Comments Character Sets	Note 1
Alternative Representation Character Sets	Note 1
Page Dimensions	-- any value
Medium Types	-- any value
Layout Path	-- any value
Layout Texture	-- any value
Protection	-- any value

Block Alignment	-- any value
Fill Order	-- any value
Coding Attributes	-- any value
Presentation Attributes	-- any value
Unit Scaling	-- any value
Fonts List	-- any value

Note 1: The default value is the minimum subrepertoire of ISO 6937/2. The only permissible values are ISO 6937/2 and ISO 8859/1.

12.8.3 Document Management Attributes

Required Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Title	-- any value

Permitted Attributes

<u>Attribute Name</u>	<u>Value Description</u>
Subject	-- any value
Document Type	-- any value
Abstract	-- any value
Document Date and Time	-- any value
Creation Date and Time	-- any value
Local Filing Date and Time	-- any value
Expiry Date and Time	-- any value
Start Date and Time	-- any value
Purge Date and Time	-- any value
Release Date and Time	-- any value
Revision History	-- any value
Organizations	-- any value
Preparers	-- any value
Owners	-- any value
Authors	-- any value
Copyright Information	-- any value
Copyright Dates	-- any value
Status	-- any value
User Specific Codes	-- any value
Distribution List	-- any value
References to Other Documents	-- any value
Superseded Documents	-- any value
Keywords	-- any value
Document Reference	-- any value
Local File Reference	-- any value
Document Size	-- any value
Number of Pages	-- any value
Languages	-- any value
Authorization	-- any value

Security Classification	-- any value
Access Rights	-- any value
Encryption Indicator	-- any value
Password	-- any value
Additional Information	-- any value

12.9. Interchange Format

The aspects of this Implementation Agreement that are concerned with the Format of the Interchange of documents are defined in this clause. These aspects include the data stream, the interchange data units, and ASN.1 encodings.

12.9.1. Data Stream

The data stream rules are according to the Interchange Format Class A, as defined in clause 4 of ISO 8613-5.

12.9.2. Interchange Data Unit Ordering

The order of interchange data units composing a document data stream must appear, when present, in the same order as that shown in Table 12.9-1, from top to bottom. For example, a Layout Object Class Descriptor interchange data unit, when present, must follow the Document Profile Descriptor interchange data unit and precede all other interchange data unit types.

Table 12.9-1 Order of Interchange Data Units

<u>i</u> Interchange Data Unit	<u>i</u> Order	<u>i</u>
<u>i</u> Document Profile Descriptor	1	<u>i</u>
<u>i</u> Layout Object Class Descriptors	2	<u>i</u>
<u>i</u> Logical Object Class Descriptors	3	<u>i</u>
<u>i</u> Text Units Representing Generic Content Portions	4	<u>i</u>
<u>i</u> Presentation Style Descriptors	5	<u>i</u>
<u>i</u> Layout Style Descriptors	6	<u>i</u>
<u>i</u> Layout Object Descriptors	7	<u>i</u>
<u>i</u> Logical Object Descriptors	8	<u>i</u>
<u>i</u> Text Units Representing Specific Content Portions	9	<u>i</u>

12.9.3. ASN.1 Generation and Parsing

This clause covers two distinct aspects of ASN.1 generation and parsing. The first aspect covers ASN.1 practices that are mandatory for an implementation to be conforming to this Implementors Agreement. The second aspect covers ASN.1 practices that are recommended by this Implementors Agreement. These recommended practices are not mandatory for conformance, but are recommended solely in the spirit of improving interoperability among different implementations.

12.9.3.1. ASN.1 Generation Requirements

There are no additional requirements, beyond ISO 8824 and ISO 8825, imposed on the ASN.1 generation.

12.9.3.2. ASN.1 Parsing Requirements

There are no additional requirements, beyond ISO 8824 and ISO 8825, imposed on the ASN.1 parsing. The treatment of ASN.1 syntax and semantic violations is at the discretion of the implementation.

12.9.3.3. ASN.1 Generation Recommendations

The focus of the ASN.1 generation recommendations is to generate ASN.1 encodings that will allow parsing by the most rudimentary of implementations. These recommendations are described in the following sub-clauses.

12.9.3.3.1. Segmenting Strings

ISO 8825 allows Bit Strings, Octet Strings, and Character Set Strings to be encoded in the Primitive form or in the Constructed form. The choice of which form to use is an option of the encoder. Using the constructed form allows a string to be segmented into a sequence of strings. This sequence of strings is then contained in the constructed form of the string. The constructed form is allowed the use of the indefinite form on content length.

This Implementors Agreement recommends that implementations limit the encoding to one level of the constructed form for Bit Strings, Octet Strings, and Character Set Strings.

For example, if of type OCTET STRING, the value '432E436F6D6273'H can be encoded in the primitive form as:

Octet String	Length	Contents
04 ₁₆	07 ₁₆	432E436F6D6273 ₁₆

The same value may be encoded in the constructed form as:

```

Octet
String Length Contents
2416 8016
      Octet
      String Length Contents
      0416 0216 432E16
      0416 0516 436F6D627316
      EOC Length
      0016 0016

```

The same value encoded using two levels of constructed form is not recommended by this Implementors Agreement. An example of an encoding containing two levels of construction is:

```

Octet
String Length Contents
2416 8016
      Octet
      String Length Contents
      2416 0416
            Octet
            String Length Contents
            0416 0216 432E16
            0416 0516 436F6D627316
            EOC Length
            0016 0016

```

12.9.3.3.2. Length Expression

ISO 8825 allows the content length of an encoding that could be expressed using the short form to also be expressed using the long form. For example, a length of one could be expressed in the short form as 00000001₂ or in the long form as 10000001₂ 00000001₂. CCITT Recommendation X.409 (1984) does not allow the same liberty in expressing the length of the encoding length. Implementations using these X.409 rules could present interoperability constraints.

This Implementors Agreement recommends that implementations generate content lengths only in their most economical form.

12.9.3.3.3. Ordering of Set Members

ISO 8824 defines sets to be unordered lists of values. It is the generator's option to select an order for the values of the set. Since this ordering is unpredictable from one implementation to the next, it is recommended that generators order the values in a set according to the order in which the members appear in the definition of the set. The intent of this recommendation is to reduce the possible interoperability problems associated with the unpredictable ordering of members in a set.

12.9.3.4. ASN.1 Parsing Recommendations

The overall intent of these parsing recommendations is to allow a high tolerance in the representation of the ASN.1 syntax without jeopardizing the semantics of the information being conveyed. Each of these tolerances is described in a following sub-clause.

12.9.3.4.1. Segmented Strings

The ASN.1 generation restriction on segmenting strings (12.9.3.3.1) is a recommendation of this Implementors Agreement and is not a requirement of ISO 8825. Therefore, it is recommended that implementations accept string encodings which have been segmented into more than one level of the constructed form.

12.10 Relationship to Other DAPs

12.10.1 SPAG

There are three Document Application Profiles (DAPs) being defined by the European Standards Promotion and Applications Group (SPAG). These are called Q/111, Q/112, and Q/113.

Q/111 and Q/112 are consistent with the NBS DAP but have limits on the attributes, particularly for logical objects, layout objects, and content types. However, even with these restrictions, Q/111 and Q/112 data streams will be subsets of the NBS DAP data streams and Q/113 is expected to be a functional equivalent to the NBS DAP.

12.10.2 CCITT

Several activities in CCITT Study Group VIII will result in application profiles being published as 1988 Recommendations. Some of this work is contained in the drafts of new and revised Recommendations, e.g., the T.400 series.

12.10.3 TOP

This NBS DAP will be presented to TOP as a suggested replacement for the TOP Version 3.0 ODA Application Profile.

12.10.4 POSI

A request for liaison has been made to POSI in order to identify Japanese-defined DAPs.

DATA MANAGEMENT

CALS REPORT ON DATA MANAGEMENT STANDARDS

SUPPORTING LOGISTIC SUPPORT ANALYSIS (LSA) USING THE
INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS)

CALS REPORT
ON DATA MANAGEMENT STANDARDS

Apr 3, 1987

(Revised)

TABLE OF CONTENTS

PURPOSE	1
BACKGROUND	1
INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS)	2
Introduction	2
IRDS Features in Support of CALS	2
Using IRDS for Logistics Support Analysis	4
IRDS Enhancements required for CALS	6
IRDS Control for Integrating Text, Graphics, and Alphanumeric Data	6
IRDS in Distributed Data Environment	6
IRDS Information and Data Modeling	8
IRDS Data Management Support	9
IRDS Graphical Input/Output	10
IRDS Life Cycle/Configuration Management	10
Recommendations for IRDS	11
DATABASE AND DATA INTERCHANGE	11
Introduction	11
Database Languages	12
Database Language NDL	12
Database Language SQL	12
CALS Application of SQL	14
Remote Database Access Services and Protocol (RDA)	15
CALS Application of RDA	15
Data Interchange	16
Current Data Interchange Standards Activities	16
CALS Application of Data Interchange Standards	17
Recommendations for Database and Data Interchange	17
SUMMARY	18

DATA MANAGEMENT STANDARDS

PURPOSE

This report is a deliverable for the FY 87 SOW on the recommendation and development of enhancements for data management standards. Specifically, the task (2.2.1.4) states:

Assess the need to enhance or tailor the NDL, SQL, IRDS standards to meet CALS objectives. The NDL and SQL were approved as ANSI standards in 1986 and the IRDS is expected to be approved in early 1988.

Following the background section, the report is divided into two sections, one for each subtask under the above task. These two subtasks are:

Possible enhancements for the IRDS include support for distributed data, data modeling, data element validation procedures, IRDS graphical input/output, and Information Systems Life-Cycle and Configuration Management. Rough draft of recommendations 3/87, enhancements to standard continuing through FY-88. (2.2.1.4.1)

Work is already under way to enhance SQL by adding features to improve data integrity, to expand ways for relating records, and to add date/time data types. Other enhancements required to support CALS will be analyzed and reported. Rough draft of recommendations 3/87, enhancements to standard continuing through FY-88. (2.2.1.4.2)

BACKGROUND

In FY 1986, NBS prepared the Preliminary Report on Data Management Standards, dated June 20, 1986, which described existing data management standards and where they could be used in existing CALS applications. The report addressed four critical areas for data management standards: (1) data structures and languages; (2) dictionaries for managing and controlling complex data descriptions; (3) data interchange; and (4) distributed data environment. In each of these areas, the report described the general use, content, and status of the appropriate standards. For data structures and languages, there are two ISO, ANSI, and FIPS approved standards: Database Language SQL, FIPS PUB 127, ANSI X3.135-1986, and ISO 9075-1987; and Database Language NDL, FIPS PUB 126, ANSI X3.133-1986, and ISO 8907-1987. For data dictionaries there are draft specifications for an

Information Resource Dictionary System (IRDS) being reviewed by ISO and ANSI. For data interchange, there are two standards: Data Descriptive File (DDF), ANSI/ISO 8211; and Abstract Syntax Notation One (ASN.1), ISO 8824/8825. Work is just beginning on standards to support the distributed data environment.

The IRDS will serve as the primary software tool within the CALS Framework Control Architecture to help manage, model, integrate, and control the CALS information environment. The use of standard database management systems (DBMS's) will provide common data access and management tools needed to support CALS. Data interchange standards (DDF and ASN.1) provide a mechanism for data structures, structured databases and files, to be easily moved from one computer system to another, independent of vendor or data model.

INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS)

Introduction

This section of the report specifically addresses subtask 2.2.1.4.1 which focuses on: (1) the IRDS features in support of CALS; (2) the use of the IRDS to support the LSA application area; (3) the IRDS enhancements that are required to support CALS; and (4) recommendations for future actions.

The IRDS Specifications have completed both the ANSI and FIPS review processes and have been approved by the American Standards Committee (ASC)/X3 Technical Committee for IRDS (X3H4). NBS anticipates that the IRDS will become an ANSI standard and a FIPS in early 1988. It is also being reviewed by ISO TC97/SC21 as a proposed Draft International Standard (DIS).

IRDS Features in Support of CALS

The IRDS Specifications were developed by NBS in cooperation with the X3H4 Committee and with important input from DoD attendees at IRDS user and vendor workshops sponsored by NBS. The IRDS was designed to be a powerful and flexible tool for managing an organization's total information processing environment. To provide for IRDS flexibility and procurement cost-effectiveness, a modularized approach was adopted. The proposed IRDS is therefore organized as a "Core" dictionary system Module and five (currently) additional optional Modules.

The IRDS Specifications include an optional module, Basic Functional Schema, which supports most organizations' requirements for documenting and managing data about the

information processing environment. This Schema defines a "starter set" of the more commonly needed types of information used by organizations to describe most existing and planned manual and automated systems. The Schema establishes the controls in the IRDS for documenting three types of information about an organization: (1) entities to document things, places, or events about the organization, e.g. documents, files, records, elements, systems, programs, modules, and users; (2) relationships which identify virtually all the connections or relationships between the entities that might prove useful to most organizations most of the time, for example, contains is used to identify that a record contains an element; and (3) attributes which describe the characteristics of entities and relationships, for example, description, length, and data type are required to document element. The Schema provides the facility for the organization to document all occurrences of the entities, relationships, and attributes.

It is not feasible for the IRDS Specifications to identify all of the types of information that might be useful to every organization in managing its information processing environment. With the "starter set," DoD will have the capability to document most of their requirements. However, in those cases where this "starter set" doesn't allow DoD to document their information requirements, the IRDS Specifications have a feature, IRDS Extensibility, which provides the capability to customize the IRDS Schema to the users environment. An example of extending the IRDS, would be to add a type of entity called SITE (along with the necessary types of attributes and relationships associated with SITE) providing DoD organizations the capability to document data in a distributed environment.

Another feature of the IRDS Specifications which can be optionally acquired is the Application Program Interface. This feature provides an interface through which IRDS commands and the resulting output can be passed between the IRDS and any programming languages having a CALL feature. With this feature, DoD can develop software to use the IRDS data for special purposes and to integrate the IRDS with their information processing systems. For example, the Joint Services' Logistics Support Analysis ADP System can be programmed to interface directly with the IRDS therefore eliminating any manual intervention that would otherwise be required.

The IRDS Specifications also provide the user with flexibility in generating output from the IRDS. One specific feature aiding the user tremendously in preparing reports and queries is a capability for the user to develop and retain lists of entities.

These entity lists allow users to easily customize reports and queries to their specific needs. This standard feature will provide DoD organizations and users with the flexibility to use the IRDS contents in a manner suitable to their CALS environment.

Because of these features, enhancements to the IRDS Specifications will be minimized. However, there are specific areas where the IRDS Specifications do need to be enhanced. These enhancements would be additional modules that will supplement the current standard, thereby expanding its functionality. In subsequent paragraphs, the envisioned enhancements to the IRDS Specifications as well as some of the Schema extensions required to support DoD CALS requirements are addressed.

Using IRDS for Logistics Support Analysis

Logistic Support Analysis (LSA) is a prime CALS application area, specifically the Logistics Support Analysis Record (LSAR), which can benefit from the use of an IRDS without enhancements to the specifications. LSAR is heavily oriented to a data dictionary environment. MIL-STD-1388-2A, Appendix F, documents the LSAR Data Element Dictionary; Appendix C documents the LSAR Master Files. The Dictionary contains all the data elements appearing in the LSAR Master Files and documents the definition, field lengths, acronyms, abbreviations, and data validation criteria for each data element.

The baseline software tool for producing an LSAR is the Joint Services' LSAR ADP System. Typically, what has happened in most batch oriented systems that have evolved over the years is that the data validation criteria for data elements are embedded within the programming language, e.g. COBOL. Programmers must manually ensure that the data validation criteria specified in the data dictionary are enforced by the programs. This process not only requires considerable manpower resources but also increases the potential for errors in the data validation procedures.

The first requirement for CALS in LSAR should be to use an automated data dictionary, an IRDS, to document and maintain the LSAR Master File Descriptions (MIL-STD-1388-2A, Appendix C) and the Data Element Dictionary (Appendix F). This will provide the LSAR users with a standard automated methodology for obtaining LSAR Master file and related data element information. More importantly, the IRDS will provide an automated tool to aid in validating the consistency and compatibility between the Master File record formats and the associated data element

specifications. The IRDS "Core" Specifications along with the optional module, Basic Functional Schema, should satisfy this LSAR requirement.

Another potential benefit of using the IRDS to support LSAR is in an automated interface to contractors' LSAR ADP systems. If the contractors are also using IRDS's, the software used to prepare the data for interchange (tape, bulk media, etc.) could use the common descriptions found in each of the IRDS's. This would help ensure that the sending and receiving systems have compatible master file and data element descriptions. One feature in the "Core" IRDS that will significantly aid in this validation process is the IRD-IRD Interface. This feature provides a controlled means for moving data from one IRDS to another even in the case where the two standard IRDS's were developed by different vendors and are resident on different hardware systems at different locations. In this latter case, it is assumed that either a communications link exists between the two computer systems or that some other means of physically moving the data is employed. Before data can be transferred, certain compatibility checks are made between the Schemas of the two IRDS's. Some differences between the two may not be significant; however, in those cases where incompatibility would result in significant problems, the data transfer will not occur. For example, if a specific type of entity in the Schema of the sending IRDS does not exist in the Schema of the receiving IRDS, any entities of that type can not be stored in the receiving IRDS. In this case, the data would not be transferred. Any compatibility problems between the two Schemas are identified and the requester is notified. Within CALS, the two IRDS Schemas can be automatically verified for compatibility before interchanging descriptions of the LSAR Master File and the Data Element Dictionary. Additionally, any time there are changes to the descriptions of the Master File records or related data elements, these changes can be made in the respective IRDS's through the IRD-IRD Interface. Not only can the changes be verified more easily, but also many changes, especially for data element validation criteria, will never require a change to software.

The IRDS can also help integrate or interface the LSAR with other DoD ADP systems, such as provisioning, packaging, and Defense Logistics Services Center (DLSC) screening systems. The use of an IRDS in each of these systems to describe the database environment, including the data elements, will provide standard procedures for determining the commonality and the compatibility between them. In addition to the IRDS "Core," the optional modules, Basic Functional Schema and Application Program Interface will need to be acquired.

IRDS Enhancements required for CALS

IRDS enhancements required to support CALS include the following: (1) control for integrating text, graphics, and alphanumeric data; (2) distributed data environment; (3) information and data modeling; (4) data management; (5) graphics input/output; and (6) life cycle and configuration management.

IRDS Control for Integrating Text, Graphics, and Alphanumeric Data

In CALS, there are four technical views depicting four basic automation capabilities: Communications, Image, Text, and Alphanumeric. The IRDS support in the communications area is discussed below under the distributed data environment. The image view is primarily a graphics representation of a part or component of a weapon system. The text view is considered to be long textual material such as a technical manual. The alphanumeric view is highly structured data that is found in a parts or LSAR database.

In CALS, there is a requirement to integrate the data from the three different views (text, graphics, alphanumeric) resulting from different processing environments. For example, constructing a technical manual requires textual procedures interspersed with graphics display of equipment components and tables of parts lists (the alphanumeric data). The IRDS can be a controlling mechanism that allows this integration of data into a single document. Using the IRDS to contain the data necessary to integrate this operation can also facilitate on-line interactive processing. For example, future implementations of Technical Information Delivery, Maintenance, and Diagnostic Aid Systems used for maintaining weapon systems would have the data necessary to integrate all three views of data.

It is possible that these DcD requirements can be satisfied by using the IRDS Extensibility feature and the Application Program Interface; however, enhancements to the IRDS Specifications could also be required. Further analysis is needed in this area to determine the specific IRDS requirements needed to support CALS.

IRDS in Distributed Data Environment

A long-range objective of CALS could be to access data from various nodes of a distributed environment. In other words, a user logged onto a system using a remote terminal could first determine what data is available and could secondly access the

data without being concerned about where that data is located or the characteristics of that data. For example, LSAR data is needed by many different users at various locations.

The existing IRDS will provide the CALS users with the basic knowledge to determine what data is available and where it is located. Additional IRDS specifications should include enhancements to support the Government Open System Interconnection Procurement (GOSIP) specification and the associated network directory functions for each layer. These facilities will document the network nodes, the data located at each node, and the dependencies between processes and data. These IRDS enhancements will document how a logical data model is distributed across multiple sites. The IRDS enhancement will also support or help support all traffic management within the network.

Other special features might include:

Mappings between database structures and mappings among database languages when the distributed data environment allows processing across heterogeneous database systems to accommodate existing non-standard databases.

Scheduling information with regard to query and application processing in order to minimize contention when updating shared data, to batch transactions, and to level the processing load.

NBS has identified some specific IRDS enhancements that are required to support distributed data. The IRDS Schema should be extended to allow information about a node (SITE) to be defined. As a minimum, the attributes that will be required include the following: address of site, processing capability at the site, permissions allowed for a site, and database information available and related controls for access and updating the data. Although the IRDS specifications do not need enhancing to document this information, any additional functionality that will be required to efficiently support the user in an interactive mode should be researched and evaluated.

Another important consideration is the method for distributing CALS data across sites. If the data is viewed as logical tables that may or may not be related to each other, then a mechanism must be built into the IRDS to set up controls for accessing and updating the distributed data. The attributes needed for the IRDS to control this distributed environment need to be

identified. For each table in a database, the following must be considered:

For a fragmented portion of a logical table, the IRDS must identify the placement of the fragmented portions and whether the logical table is partitioned or replicated. A partitioned table is broken into pieces and the pieces stored at various nodes of the network. A replicated table has duplicate copies located at various nodes in the network.

If the logical table is replicated, then the master site for each portion must be identified and updates to the data at the master site must also be made in every replicated copy throughout the network. If a new site is designated the master, then this fact must be properly reflected in the data at each site.

If the data is partitioned, then updates normally need to occur at only one site. However, if the data is moved from one site to another, then the data must be inserted at the new site and deleted from the old site.

For a partitioned logical table, is it horizontally partitioned (different rows are at different sites) or vertically partitioned (different columns are at different sites)? Horizontal partitioning will mean that a query may need to be sent to multiple sites to obtain the complete answer. Vertical partitioning will cause tables residing at different sites to be joined to form one logical table.

IRDS Information and Data Modeling

The CALS Framework specifies a requirement to model the CALS information environment (Information Architecture); a specific methodology is not recommended. Some of the commercially available data dictionary systems do support information modeling but no standard specifications have been developed. Logically, the IRDS is the tool that should be used for documenting the CALS information environment. To do this, the IRDS can be extended to include the necessary entities, attributes and relationships. Also, the IRDS specifications should be enhanced to provide the functionality necessary to evaluate the relationships between the information descriptions in the IRDS, for example, evaluate the inter-record relationships within a data model or database.

Some specific requirements have been documented for enhancing the IRDS. A designed or proposed data model needs to document

certain restrictions on relationships. For example, is a relationship 1 to 1, 1 to many, or many to many? Is the table an independent entity or a dependent entity (parent-child relationship)? If the implementation schema does not support the concept of a domain, the IRDS must identify columns that can be meaningfully compared. There is a need for supertypes/subtypes. From one perspective objects have the same characteristics and are managed in the same way; from another perspective there are distinct characteristics. The most common solution to this problem is to join all characteristics allowing some attributes to be null. Application code is written to check which kind of object it is and test to see that the needed attributes are provided. An example of this is an inventory application that manages parts and tools as the same kind of object and a tool crib application which has additional attributes about tools. Certain constraints are also useful and are described in SQL Data Integrity in the section on database enhancements.

IRDS Data Management Support

There is an additional IRDS enhancement needed to help the data administrator in supporting: (1) the standardization of data elements; and (2) a "thesaurus" capability. It will also help support the use of the IRDS in LSAR and the distributed data environment.

The support for data element standardization must occur throughout the life cycle of a data element. Once a data element is identified during the initial phases, facilities will be required to assure that:

The name associated with the data element is used consistently throughout the life-cycle. For example, when a data element is referred to in a database, only the standard name applicable to that environment should be used.

Characteristics, for example, character strings, fixed-point numbers, floating-point numbers, for the data element correspond to the intended use of that data element.

Validation criteria associated with the standard data element and the variety of usage environments for the standard data element must be controlled and available to the facilities that perform validation. For example, when a user enters data on a data entry screen, the IRDS should contain the criteria for validating the users input before the data is entered into a database. The validation

criteria may be a range of numbers, list of values, or reference to a table containing valid values.

The "thesaurus" capability will aid the user in locating and accessing the data even though the appropriate name for the data is not known. This capability will require increased support for indexing or classifying the data through the use of a series of keywords. For example, keywords for Finance-Department might be Accounting and Payroll. A user could locate the Finance-Department data by specifying either Accounting or Payroll. Also, this capability will help the person responsible for data element standardization resolve synonym and homonym problems.

IRDS Graphical Input/Output

The IRDS currently contains two user interfaces: a Command Language and a Panel Interface. The Panel Interface may be considered "user-friendly," in that it leads the user through the appropriate panels (or logical screens) to accomplish the desired function. The results would be the same as if the user entered a command through the Command Language Interface.

A powerful new means of communicating with the IRDS could be built around an interface to graphics terminals and systems. Using such an interface, users could draw diagrams (data modeling, process flow, etc.) for input to the IRDS and could see the answers to their queries displayed in the form of these diagrams. This enhancement to the IRDS will also help integrate CALS text, graphics, and alphanumeric data.

IRDS Life Cycle/Configuration Management

In CALS, emphasis is being placed on Life Cycle and Configuration Management of weapon systems. The same considerations need to also be given to the acquisition and operation of information processing systems supporting CALS. The current IRDS Specifications have identified fairly extensive requirements for the management of information systems life cycle phases. Further enhancements to the IRDS might include specifications for integrating the life cycle phases with the facility that checks the quality of the entities in the IRDS. This could help in determining the "suitability" of moving entities to another phase. This enhancement would also include a facility to:

Establish and manage configurations (i.e., treating assemblages of processes and data as a structure).

Establish baselines associated with life cycle phases, and rules to control movement across these baselines, in both directions.

Recommendations for IRDS

For the next NBS deliverable on the Strategy for Using the IRDS in CALS, NBS could do one or more of the following:

Using a subset of data (a specific part) from an actual weapon system, illustrate how the IRDS would be used to:

document the LSAR Data Element Dictionary.

interface with another ADP LSAR system also using a standard IRDS,

standardize common CALS data elements, e.g. LSAR data elements, and

control the integration of text, graphics, and alphanumeric data.

Illustrate how the IRDS would be used to support the complete Life Cycle and Configuration Management of information systems during the acquisition and operation process.

Illustrate the use of the IRDS to model data in a technical manufacturing environment.

DATABASE AND DATA INTERCHANGE

Introduction

This section of the report specifically addresses subtask 2.2.1.4.2 on the possible enhancements for the database languages and data interchange standards.

The last year has been very productive in the area of database standards. Database Language NDL became an ANSI standard (ANSI X3.133-1986) on August 1, 1986. Database Language SQL (ANSI X3.135-1986) followed soon, on October 16, 1986. These standards provide formal specifications for schema definition and data manipulation language of two popular data models. Considerable effort was expended to promote identical database standards

within ANSI and ISO. Early in 1987, documents identical to the ANSI standards (ISO 8907 for NDL and ISO 9075 for SQL) were approved by ISO TC 97 member body ballot. And then on March 10, 1987 both NDL and SQL were adopted as Federal Information Processing Standards, FIPS PUBS 126 and 127, respectively.

Database Languages

In the area of database languages, the following paragraphs discuss the enhancements to the NDL and SQL standards, CALS application of SQL, Remote Database Access Services and Protocol (RDA), and CALS application of RDA.

Database Language NDL

Database language NDL, for the network data model, is suited for use in highly structured applications requiring rapid access along predefined paths. Work on this standard is essentially complete; however, NBS is sponsoring a separate companion standard for an embedded-syntax style of interface between a programming language and the database management system. The programming languages Ada and C will be included. Originally, only COBOL, FORTRAN, Pascal, and PL/I were supported.

Although the network data model has proven successful for many existing applications, it does not currently enjoy the popularity of the relational data model. Software vendors are more interested in investing their limited development resources in relational products, and consequently, there is little interest in further development of the NDL language. Nevertheless, the NDL standard is expected to benefit those Federal agencies which have existing network model databases and need to replace existing hardware and software. In cases where it is neither appropriate nor cost effective to use the relational model, the NDL standard can be used in the procurement of new and replacement computer systems.

Database Language SQL

Database Language SQL, for the relational data model, is appropriate for applications requiring flexibility in the data structures and access paths of the database. The relational data model is desirable where there is a substantial need for ad hoc data manipulation by end users who are not computer professionals, in addition to the need for access by applications under production control. Standards development for SQL is very active, involving four projects for the ASC/X3 technical committee on database, X3H2. These projects are: 1) an

addendum to SQL to provide a data integrity enhancement feature; 2) an embedded-syntax style of interface between the DBMS and any of the programming languages Ada, C, COBOL, FORTRAN, Pascal, and PL/I; 3) extended SQL (SQL2); and 4) Remote Database Access Services and Protocol.

The first two projects are limited in scope, rounding out the basic SQL standard. The other two projects represent a window of opportunity for CALS requirements to influence the development of database software over the next seven years.

It is important to note that this window of opportunity requires immediate action; the next extension of SQL is not expected for another seven years. X3H2 had planned to extend SQL soon after the base standard was approved, and indeed this promise was made at the time of the SQL public review in response to criticism that the proposed SQL standard needed additional functionality; therefore, X3H2 plans to close the document to additional features by year end.

Extended SQL will add to the basic functionality of SQL; including such features as full referential integrity, domains, user defined data types, outer join, date/time data types, bit (octet or byte) data type, schema manipulation, schema information tables, temporary tables and views, enhanced status feedback and diagnostic tables, interactive browse, dynamic SQL, enhanced character handling, additional security features, and tree structured (recursive) data. All of these features have been identified by X3H2 as potential work items; however, progress is not made until committee members submit proposals defining the specifications to be incorporated into the extended SQL standard. Additional requirements for enhancements to SQL have already been defined by NBS. These enhancements include vector and array data types, ordered sets (lists), and uniqueness constraints enforced across several tables.

The importance of incorporating as many of these features as possible into the next version of SQL cannot be over-emphasized. These are features which are needed by a broad spectrum of CALS applications. Most vendors will provide these features, but each vendor will implement them differently. Without standardization, programs accessing databases will not be portable across computer environments. Users accessing data on various computers will need to be aware of which site they are accessing as well as the idiosyncracies of that site. This will make it more difficult to attain the goal of site transparency.

NBS has submitted proposals for the following enhancements: full referential integrity, outer join, schema information tables, and an additional security feature. These enhancements will all contribute to the CALS goals. NBS has also made numerous minor technical contributions. However, time is running out, and proposals are needed for schema manipulation, temporary tables and views, enhanced character handling, security features, and tree structured data. These proposals must be developed and promoted within ANSI and ISO.

NBS has successfully accomplished its goal of obtaining identical ANSI and ISO standards for SQL. The economic implications of identical standards are significant. The expanded international market for products conforming to the ANSI standard (and consequently to the ISO standard) reduces the risk and expands the opportunities to vendors who develop conforming SQL products. Due to the popularity of the SQL standard, many vendors are expected to implement an SQL interface. Utilizing the emerging standard for the C Programming Language, many vendors will "port" their product to a variety of hardware architectures. Users will be able to procure a uniform database environment with identical user interface across dissimilar hardware. Third-party software and applications based upon the SQL standard will greatly increase the selection and quality of off-the-shelf products available to users.

CALS Application of SQL

Of particular interest to CALS requirements is the capability of retrieving tree-structured data within SQL. This retrieval capability is especially useful in modeling configuration data, which is hierarchical in structure, and nested to varying levels. This retrieval capability could locate all components in a subsystem, even when some components are defined as being composed of several other components. Or, this retrieval capability could locate all documentation packages related to a configuration item on a ship -- when any package could itself be comprised of several other packages. Alternately, all components which require a particular documentation package could be located, even though the documentation package may be included in several unrelated documentation packages. Although X3H2 would most likely incorporate this retrieval capability into SQL2, it is unlikely that any committee member other than NBS will perform the research or write the actual proposal.

Another enhancement to SQL which would satisfy some of CALS requirements for modeling geometric data is the NNF (non-normal form) extensions to SQL which are being implemented by the

Automated Manufacturing Research Facility (AMRF) project at NBS. The AMRF is working closely with IBM Heidelberg, which already has created a working prototype of NNF. Since the membership of X3H2 represents users and vendors for business applications, a proposal for NNF extensions would be viewed as a research project, which indeed it is. However, the committee would be receptive to status briefings as research progresses. If a solid proposal can be developed, based on NBS experience, then NBS would be positioned to promote it as an optional module of SQL2.

Remote Database Access Services and Protocol (RDA)

X3H2 has just received approval for a new project to work on remote database access, in cooperation with the ISO Remote Database Access Rapporteur Group. RDA is an application layer service providing access to a shared data resource from a program which may reside in a workstation or a mainframe. RDA is dependent upon OSI application layer standards for ROS (ISO DP 9072) and CCR (ISO 8649/3), as well as Presentation (ISO 8822 and 8823), ASN.1 (ISO 8824 and 8825), and Association Control (ISO 8649/2). The communications requirements of this application are typical of a broad class of data access applications. The RDA proposal will address the requirements for association management, invoking server functions, transaction management, and bulk data transfer. RDA will be built on several other standards, notably SQL and ASN.1 (Abstract Syntax Notation). The SQL2 document has already been modified to support RDA needs for schema information tables. Further changes to SQL have been requested, such as temporary tables and asynchronous data manipulation language.

RDA is a much-needed interface between dissimilar databases and would greatly benefit many future CALS applications. The RDA standard is not based upon existing products, and as such should be prototyped to prove feasibility and completeness. NBS should assist in the development of a prototype to ensure that user requirements are adequately addressed. Rapid approval is not expected for a RDA standard; however, NBS participation could hasten the development and acceptance of this important interface.

CALS Application of RDA

The RDA standard will facilitate the interchange of data among the processes of a computer integrated logistics system. Data from a central database supports the design, planning, manufacturing, inspection, provisioning, and maintenance of products. Often these processes are run on workstations

accessing their own local databases, and need a standard protocol to request data services from the central database and to upload/download selected tables of information. Or, any given process may need to access a collection of data distributed across several workstations; e.g. integrating the electrical and mechanical subsystems designed on stand-alone systems.

On the shop floor, RDA would provide a protocol for the real-time exchange of structured data between the interacting components of an automated manufacturing facility.

The following are a few of the CALS projects which would benefit from the rapid development of an RDA standard: Integrated Design Support (IDS - to manage technical information across the entire life cycle of a weapons system), Integrated Information Support System (IISS - to service a distributed set of heterogeneous computer hardware and software systems accessible from geographically dispersed locations), Logistics Information Management Support System (LIMSS - to network the various forms of information processing via telecommunications for all levels of logistics functions), and Technical Logistics Reference Network (TLRN - to network multiple data bases of information with multiple user types to do procurement actions).

Data Interchange

Current Data Interchange Standards Activities

The ASC/X3 technical committee on data interchange, X3T2, was formed last August to perform work recommended by the SPARC Data Interchange Study Group (DISG). X3T2 has already contributed to the technical specifications of the ISO standard for ASN.1 and is working towards adopting an ANSI version of the standard. X3T2 is charged with maintenance projects for the standards for Data Descriptive File (ANSI/ISO 8211) and Representation of Numerics in Character Strings (ANSI X3.42 and ISO 6093).

Two new projects of particular interest to CALS are Common Language-Independent Data Types and Common Language-Independent Procedure Calling Mechanisms.

Common Language-Independent Data Types will facilitate exchange of data between totally separate and independent systems. This standard will also support the interchange of data in mixed language programming environments, whether in local or distributed mode.

Common Language-Independent Procedure Calling Mechanisms will increase the modularity and portability of software routines. Such a standard would allow applications written in one language (e.g. GKS, SQL, Ada, COBOL, etc.) to invoke libraries of code or functions written in other languages, both in local and remote processing. It is anticipated that there will be less need to rewrite one program in the language of another, to write a custom interface mechanism, or to restrict coding to only languages that can already communicate on a given machine.

For common language-independent data types and procedure calling mechanisms, there is hope for rapid standardization. Although there are no base documents at this point, the technical aspects of the problem are well understood. Every vendor has solved the problem in his own environment. The greatest obstacle to standardization will be the difficulty of building consensus.

CALS Application of Data Interchange Standards

In the engineering environment a CAD package written in one language may need to communicate with a subroutine written in another language. For example, a routine written in assembler or C (by the vendor) may draw components, such as piping, ducting, or wiring, while another routine, written in FORTRAN (by the vendor or user) may perform engineering analyses of the subsystem being drawn. In the logistics environment, a "user-friendly" 4GL (fourth generation language) routine, such as a data-entry application for configuration management, may need to call a COFOL routine, such as a data validation subroutine.

Common language-independent data types and procedure calling mechanisms will facilitate the integration of programs written in different languages on the same machine or programs written in the same language on different machines. Often, users are not aware that an integration problem exists, because on any given computer, the problem is solved by the vendor in a unique way; however, as the need increases for real-time interaction of dissimilar programming languages on different hardware architectures, the importance of these two standards will become obvious.

Recommendations for Database and Data Interchange

Top priority should be given to enhancing SQL functionality prior to the public review of SQL2.

NBS should develop and submit for incorporation into SQL2 a proposal for the retrieval of tree-structured data.

NBS should participate actively in the international development project for remote database access (RDA) and should assist in the development of a prototype.

NBS should continue research on the NNF (non-normal form) extensions to SQL and should brief X3H2 on their progress.

NBS should complete work on the embedded-syntax interface to NDL. This standard would provide a more convenient interface for the programming languages COBOL, FORTRAN, PL/I, and Pascal. This standard would also provide an NDL interface to the languages Ada and C.

NBS should promote rapid standardization of common language-independent data types and procedure calling mechanisms.

SUMMARY

Many of the enhancements to the IRDS, SQL, and NDL described in this report apply to any information system including those in the CALS environment. However, a few of the enhancements primarily apply to CALS such as the use of the IRDS to control integrating text, graphics, and alphanumeric data and the requirement for the SQL non-normal form for modeling geometric data. With the IRDS extensibility feature, the IRDS Standard Schema can be customized to the CALS environment without enhancements to the IRDS Specifications. Enhancements to the IRDS Specifications will be necessary only when the functionality of the IRDS must be expanded to suit CALS needs. CALS should continue to support enhancements to the IRDS and SQL standards.

DATA MANAGEMENT

REPORT

SUPPORTING

LOGISTIC SUPPORT ANALYSIS (LSA)

USING THE

INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS)

September 30, 1987

SUPPORTING
LOGISTIC SUPPORT ANALYSIS (LSA)
USING THE
INFORMATION RESOURCE DICTIONARY SYSTEM (IRDS)

Introduction

Logistic Support Analysis (LSA) is a CALS application area, specifically the Logistics Support Analysis Record (LSAR), which can benefit through the use of an Information Resource Dictionary System (IRDS). LSAR is heavily oriented to a data dictionary environment. MIL-STD-1388-2A, Appendix C documents the LSAR Master Files and Appendix F documents the LSAR Data Element Dictionary. NBS is working with the CALS Policy Office on a project to demonstrate the use of the IRDS in support of the LSAR. This report documents actions completed in FY87 and actions planned for FY88.

Background

In April 1987, NBS provided the CALS Policy Office a report on Data Management Standards, included separately in this NBS FY 1987 CALS report. It specifically addressed the recommendation and development of enhancements for data management standards. One of the recommendations in this report was for NBS to illustrate the use of an IRDS to support the LSAR environment. As a result of this recommendation, the OSD CALS Policy Office initiated the follow-on project for NBS to demonstrate the feasibility of using an IRDS in the LSAR environment. The CALS Policy Office designated the OASD Weapons Support Improvement and Analysis Office to work with NBS and coordinate the activities of this project.

Summary of Completed Actions

NBS used the MIL-STD-1388-2A, Appendix C and Appendix F, as a basis for developing the initial logical model of the LSAR environment. A subset of the LSAR data (H and H1 records) was selected because it would show some of the more complex relationships of the data that would appear in the rest of the LSAR.

With assistance from the Weapons Support Improvement and Analysis Office, an entity relationship model (E-R) of the LSAR subset data was developed. The resulting model containing the different entities, attributes and data element descriptions were entered and analyzed using JANUS, an automated tool that is based on an extended entity relationship (IDEF1X) methodology. The data model is included in Appendix A of this report and Appendix B

includes several reports generated by JANUS during the data analysis phase. The finished enterprise model was subsequently loaded into the IRDS. Some examples of the IRDS data definition language (DDL) used to create the support item identification and provisioning data model are included in Appendix C. It was necessary to extend the IRDS through the IRDS extensibility feature to provide the capability to document associate data types, valid data value ranges, and valid data value codes for the attributes. Further information on the extensions made to the IRDS is located in Appendix D. Additional extensions are being made to provide the ability to explicitly document certain relationships and constraints that need to be placed on the LSAR data model. At this point in time, there are no known enhancements that need to be made to the IRDS standard to fully document the LSAR data.

NBS developed a scheme to compare the data descriptions in the DoD Unified Data Base (UDB) data dictionary, using Cullinet's IDMS Integrated Data Dictionary (IDD), to the baseline IRDS schema. It was decided the best alternative for comparing the contents of the two dictionaries was to extract the IDD schema and translate it into IRDS Command Language statements that can be readily examined. Comparison routines were then written to compare entity type, access name, and properties of the data.

NBS has acquired and reviewed the UDB database description documentation. The UDB prototype has faithfully implemented the intent of the MIL-STD-1388-2A. However, there are subtle and useful relationships for managing data integrity and design of input/output products that are not described in the MIL-STD-1388-2A and consequently missing in the UDB. They were documented in the IRDS. When the comparison is made between the contents of IRDS and IDD dictionaries, these differences should be highlighted.

In order to compare the schema in the IRDS and the IDD through the comparison routines, the IDD output will have to be translated to the neutral baseline IRDS format. NBS will be writing the specifications for translating the IDD Schema to the IRDS Schema using the IRDS Command Language as a neutral format. Specifications for this translation facility will be completed after NBS obtains current IDMS documentation. The conversion routines which then interface with the IRDS standard are in the domain of the vendor, it will not be feasible to write the program routines to do the comparison only.

Future Work

NBS will be completing the translation specifications mentioned above. Additionally for FY88 deliverables, NBS is prepared to

demonstrate the types of differences that would occur when making the comparison. Emphasis would be placed on the more complex relationships that are documented in the IRDS but not in the IDD.

Another FY88 deliverable can also be a paper prepared by NBS which will describe to the user the areas that will be evaluated during the comparison process. This would be an abbreviated users manual and will provide the information necessary to intelligently compare the two dictionaries.

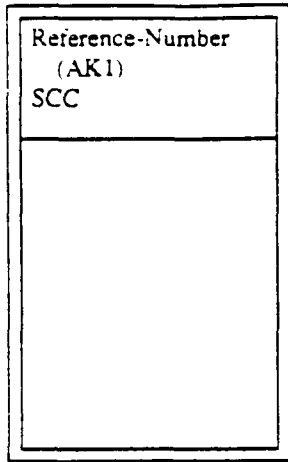
APPENDIX A - The Data Model

In addition to describing the physical or structural properties of information and where it is used, a data dictionary is an enterprise resource which describes what the data means and how it is related to other information. The meanings and relationships are the foundation of the operating procedures and automated processes which maintain the data integrity in a functioning system. A logical data model is a formal representation of these characteristics. A logical model of the LSAR was developed using an extended entity relationship methodology IDEF1X automated by the JANUS tool. IDEF1X was developed through an Air Force ICAM project and has been proposed as the CALS internal standard. The relevant data relationships and constraints were developed and reviewed with the customer. This model was the basis for our work with the IRDS.

The following pages contain the entity-relationship model of the support item identification and provisioning portions of the LSAR. To understand the data model a few basic concepts must be clear. Each of the boxes represent an entity which is a logical collection of atomic data elements or attributes. The boxes have a horizontal line which divides the properties which uniquely define an instance of the entity above the line and properties which further describe the object or concept below the line. The additional properties which would normally be found below the line are only in the detailed entity contents report in Appendix B and are under the heading of extended data. The lines that connect the entities represent the relationships between these objects or concepts. In the IDEF1X model, an entity which depends upon another is represented by a solid line from its parent entity with a big dot at the end toward the subordinate entity box. There exists only two independent entities in this data model, Item-Primary-Reference and Weapon-System; these are represented with a double lined box and are placed in the upper corners of the report. A complete description of graphic conventions are described in the legend that follows the model.

Item-Primary-Reference

is Used in



is Illustrated in Technical Manual

A

Has Requirements for

B

Has Fabrication techniques to Manufacture

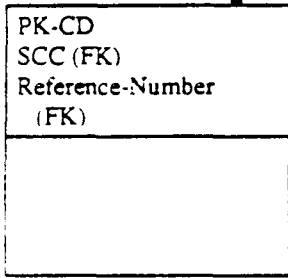
is the Form. Fit,Function Equivalent

Allowance Quantity For

is Packaging Requirements

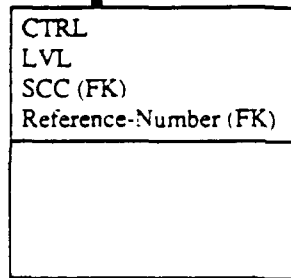
Packaging

0:3

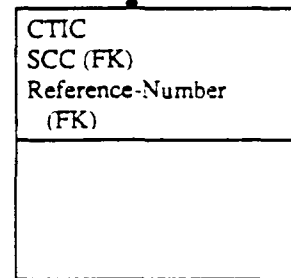


0:9

Basis-Of-Issue



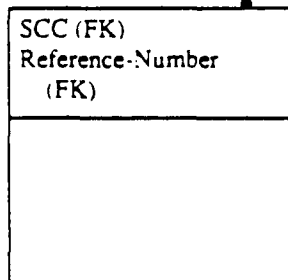
Potential-Supply-Sources



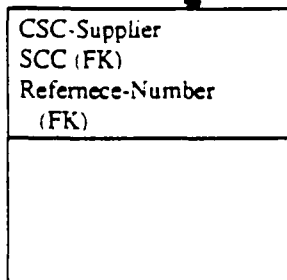
Has Current issue Price of

Block-Issue-Validation

Z

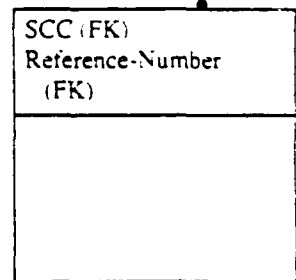


Additional-Vendors

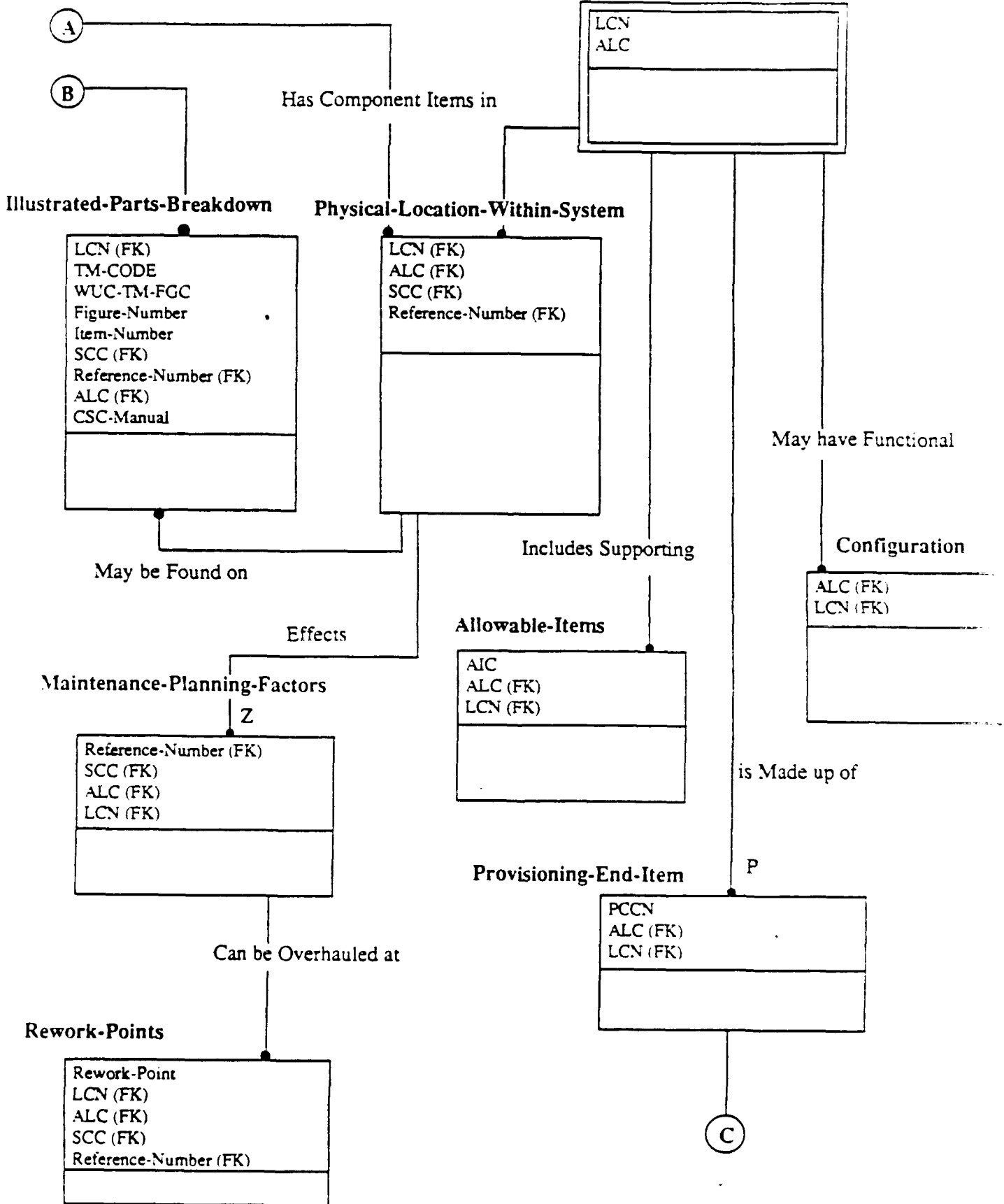


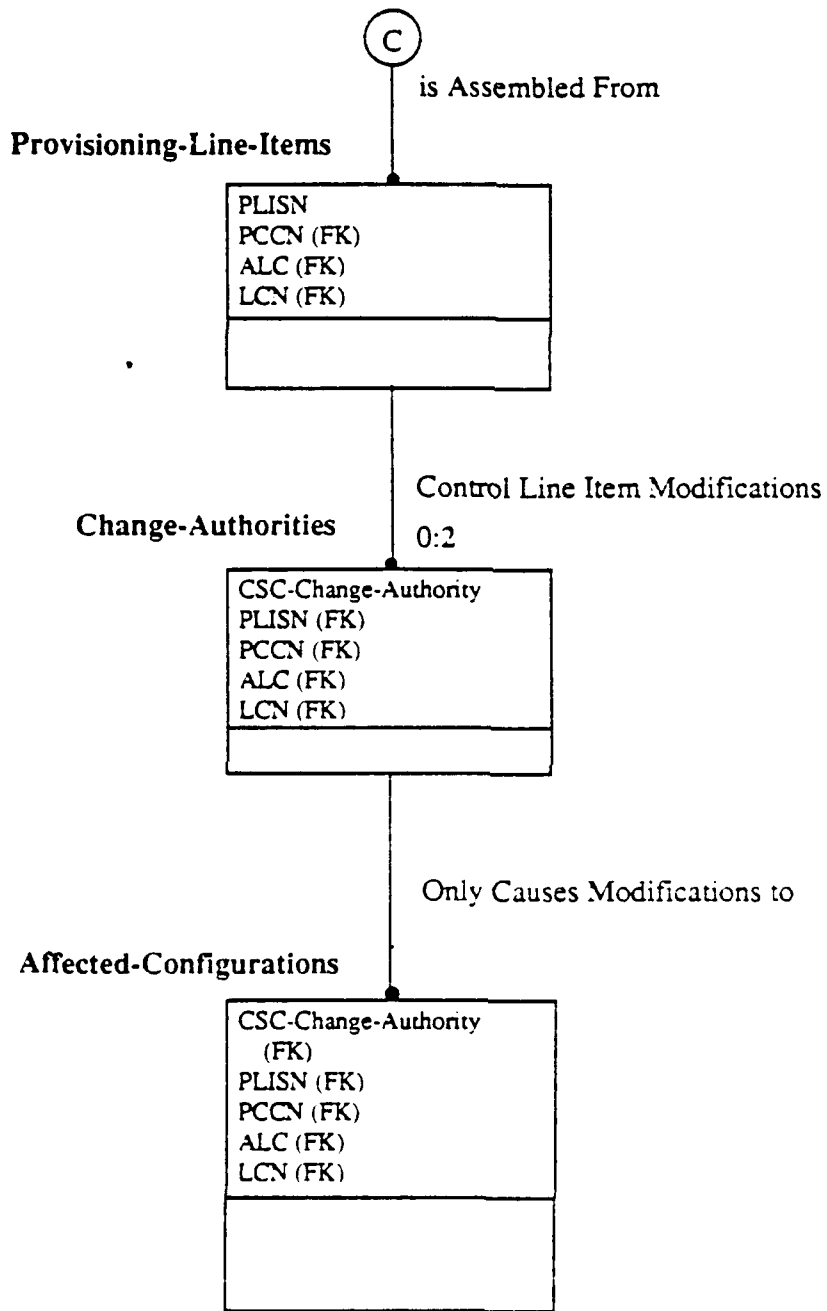
Storage-And-Disposal

Z



Weapon-System

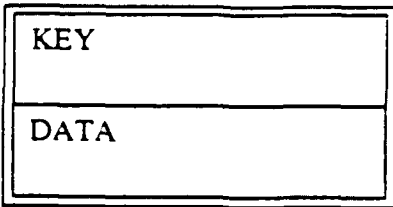




NOTE: The letters A, B and C are merely continuations of the data model from the adjacent pages.

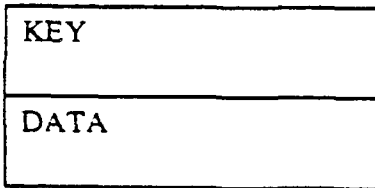
LEGEND for JANUS Data Model Graphs

ENTITY-NAME



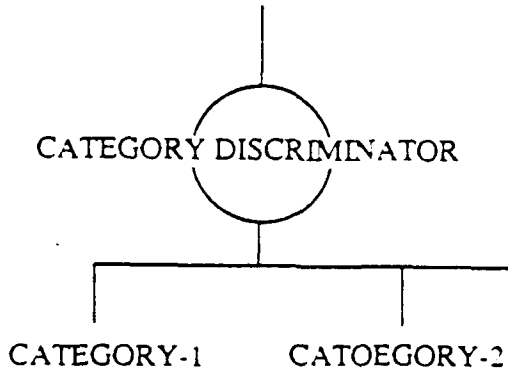
An Independent entity can be identified solely by its key.

ENTITY-NAME



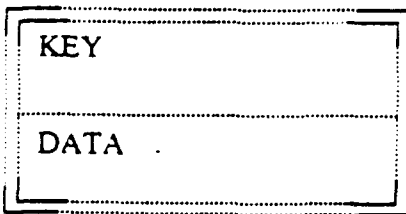
An Dependent entity requires other entities for its unique identification.

GENERIC ENTITY



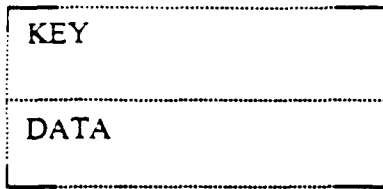
An instance of a Category exists if and only if an instance of the generic entity exists with a corresponding value of the category discriminator.

ENTITY-NAME



The shadow of an independent entity which is not selected.

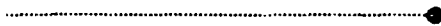
ENTITY-NAME



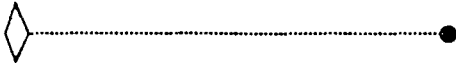
The shadow of an dependent entity which is not selected.



The key of the big dot entity includes the key of the other entity, which must exist.



The data of the big dot entity includes the key of the other entity, which must exist.



The data of the big dot entity includes the key of the other entity, which is either null or must exist.



Blank; for one instance of an entity there may be many instances of the big dot entity.



1 only; for one instance of any entity there is one and only one instance of the big dot entity.



0 or 1; for one of instance of an entity there is either one or no instance of the big dot entity



For one instance of an entity there is at least one instance of the big dot entity.

ATTRIBUTE (FK)

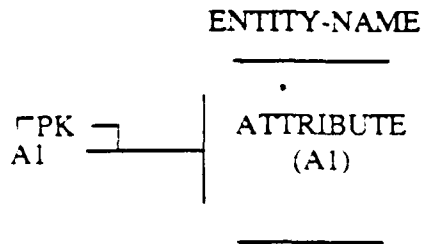
This identifies an attribute as a foreign key.

ROLE-NAME.ATTRIBUTE

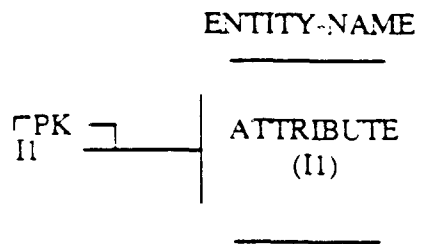
When an attribute is a role name for another attribute, it will be suffixed with the name of that attribute.

ROLE-NAME.(ATTRIBUTE-1
ATTRIBUTE-2 ...)

When an attribute is a role name for a list of attributes, it will be suffixed with the list of those attributes.



The entity is uniquely identified by Alternate Key 1, which consists of the annotated attributes.



The entity information may be obtained based on Inversion Entry 1, which consists of the annotated attributes.

APPENDIX B - JANUS Reports

JANUS is an analysis tool developed by DACOM that utilizes the entity-relationship method as extended by IDEF1X to analyze data. After completing the analysis this tool enables a variety of information that can be generated using the data model as input. The following reports are examples of this capability. One of the most useful functions is creating the BUSINESS RULES REPORT which produces English sentences that describe the relationships between data entities and the constraints such as cardinality which are needed to maintain the consistency of actual data instances. This type of report is particularly valuable for validating the results of the analysis with subject area experts who are not well versed in the modeling methodology.

The following report consists of the listing of the entities within the model.

ENTITY LISTING

ADDITIONAL-VENDOR
AFFECTED-CONFIGURATION
ALLOWANCE-ITEM
BASIS-OF-ISSUE
CHANGE-AUTHORITIES
CONFIGURATION
ILLUSTRATED-PARTS-BREAKDOWN
ITEM-PRIMARY-REFERENCE
MAINTENANCE-PLANNING-FACTOR
PACKAGING
PHYSICAL-LOCATION-WITHIN-SYSTEM
POTENTIAL-SUPPLY-SOURCE
PROCUREMENT-PRICE
PROVISIONING-END-ITEM
PROVISIONING-LINE-ITEM
REWORK-POINTS
STOCK-ISSUE-VALUATION
STORAGE-AND-DISPOSAL
WEAPON-SYSTEM

The following pages contain the ENTITY CONTENTS report which creates an entry for each entity including the glossary description of each and a list of the key attributes and the extended (non-key) data. The relationships to other entities are also listed.

ENTITY CONTENTS

*** ADDITIONAL-VENDOR ***

GLOSSARY DESCRIPTION

The list of vendors that can produce the part is included here but it must be noted that the characteristic data for the part may differ from vendor to vendor.

EXTENDED DATA

ARN
CSC-SUPPLIER
FSCM
REFERENCE-NUMBER
RNCC
RNVC
SCC

PRIMARY KEY

CSC-SUPPLIER
REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE Form,Fit,Function Equivalent
ADDITIONAL-VENDOR
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** AFFECTED-CONFIGURATION ***

GLOSSARY DESCRIPTION

List of the Usable on codes which denote the valid configurations for a line item.

EXTENDED DATA

ALC
CSC-CHANGE-AUTHORITY
LCN
PCCN
PLISN
UOC

PRIMARY KEY

ALC
CSC-CHANGE-AUTHORITY
LCN
PCCN
PLISN

IDENTIFYING RELATIONSHIP AS CHILD

CHANGE-AUTHORITIES Only Causes Modifications To
AFFECTED-CONFIGURATION

FOREIGN KEY PLISN
FOREIGN KEY LCN
FOREIGN KEY ALC
FOREIGN KEY PCCN
FOREIGN KEY CSC-CHANGE-AUTHORITY

*** ALLOWANCE-ITEM ***

GLOSSARY DESCRIPTION

Special considerations items that although not a part of an item may be ordered to embellish the functionality of a line item.

EXTENDED DATA

AIC
AIC-QTY
ALC
LCN

PRIMARY KEY

AIC
ALC
LCN

IDENTIFYING RELATIONSHIP AS CHILD

WEAPON-SYSTEM Includes Supporting ALLOWANCE-ITEM
FOREIGN KEY ALC
FOREIGN KEY LCN

*** BASIS-OF-ISSUE ***

GLOSSARY DESCRIPTION

Used only for special tools, the item is not a part of the end item.

EXTENDED DATA

CTRL
EI
LVL
QTY-AUTH
REFERENCE-NUMBER
SCC

PRIMARY KEY

CTRL
LVL
REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE Allowance Qty Defined for End
Item/System BASIS-OF-ISSUE
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** CHANGE-AUTHORITIES ***

GLOSSARY DESCRIPTION

The list of the valid authorities that can order a change in the end item.

EXTENDED DATA

ALC
CHANGE-AUTHORITY
CSC-CHANGE-AUTHORITY
FROM-SERIAL-NO
IC
LCN
PCCN
PLISN
QTY-PROCURED
REPL-SUPERCEDED-PLISN
RS-IND
TIC
TO-SERIAL-NO

PRIMARY KEY

ALC
CSC-CHANGE-AUTHORITY
LCN
PCCN
PLISN

IDENTIFYING RELATIONSHIP AS PARENT

CHANGE-AUTHORITIES Only Causes Modifications To
AFFECTED-CONFIGURATION

IDENTIFYING RELATIONSHIP AS CHILD

PROVISIONING-LINE-ITEM Control Line Item Modifications
CHANGE-AUTHORITIES

FOREIGN KEY PCCN
FOREIGN KEY ALC
FOREIGN KEY LCN
FOREIGN KEY PLISN

*** CONFIGURATION ***

GLOSSARY DESCRIPTION

Deals with functionality of end item.

EXTENDED DATA

ALC
LCN
UOC

PRIMARY KEY

ALC
LCN

IDENTIFYING RELATIONSHIP AS CHILD

WEAPON-SYSTEM May Have Functional CONFIGURATION
FOREIGN KEY ALC
FOREIGN KEY LCN

*** ILLUSTRATED-PARTS-BREAKDOWN ***

GLOSSARY DESCRIPTION

All information pertinent to manuals which contain a particular item.

EXTENDED DATA

ALC
CSC-MANUAL
FIGURE-NUMBER
ITEM-NUMBER
LCN
PROV-NOMEN
QTY-FIG
REFERENCE-NUMBER
SCC
TM-CHG-NO
TM-CODE
TM-IND
TMI
WUC-TM-FGC

PRIMARY KEY

ALC
CSC-MANUAL
FIGURE-NUMBER
ITEM-NUMBER
LCN
REFERENCE-NUMBER
SCC
TM-CODE
WUC-TM-FGC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE is Illustrated in Tech Manual
ILLUSTRATED-PARTS-BREAKDOWN
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER
PHYSICAL-LOCATION-WITHIN-SYSTEM May be Found on
ILLUSTRATED-PARTS-BREAKDOWN
REFERENCE-NUMBER
FOREIGN KEY SCC
FOREIGN KEY ALC
FOREIGN KEY LCN

*** ITEM-PRIMARY-REFERENCE ***

GLOSSARY DESCRIPTION

The main information about an item from the LSAR perspective including identifying fields and some physical characteristics.

EXTENDED DATA

CTIC	NSN
DAC	PLCC
DSR-R	PLT
FSCM	PPSL
HCI	PTD-SELECT
ICC	REF-NUM-OVERFLOW
IMC	REFERENCE-NUMBER
ITEM-NAME	RNCC
MAC	SCC
MAOT	SIC

PRIMARY KEY

REFERENCE-NUMBER
SCC

ALTERNATE KEYS

ALTERNATE KEY 1
REFERENCE-NUMBER, FSCM

IDENTIFYING RELATIONSHIP AS PARENT

ITEM-PRIMARY-REFERENCE Form,Fit,Function Equivalent
ADDITIONAL-VENDOR
ITEM-PRIMARY-REFERENCE Allowance Qty Defined for End
Item/System BASIS-OF-ISSUE
ITEM-PRIMARY-REFERENCE is Illustrated in Tech Manual
ILLUSTRATED-PARTS-BREAKDOWN
ITEM-PRIMARY-REFERENCE Has Packaging Requirements PACKAGING
ITEM-PRIMARY-REFERENCE is Used in
PHYSICAL-LOCATION-WITHIN-SYSTEM
ITEM-PRIMARY-REFERENCE Has Fabrication Techniques to
Manufacture POTENTIAL-SUPPLY-SOURCE
ITEM-PRIMARY-REFERENCE Has Econ lot size Determined By
PROCUREMENT-PRICE
ITEM-PRIMARY-REFERENCE Has Current Issue Price of
STOCK-ISSUE-VALUATION RATIO Z
ITEM-PRIMARY-REFERENCE Has Requirements For
STORAGE-AND-DISPOSAL RATIO Z

*** MAINTENANCE-PLANNING-FACTOR ***

GLOSSARY DESCRIPTION

Information pertinent to the maintenance cycle of an item including replacement, repair procedures and item expected life.

EXTENDED DATA

ALC
DMIL
LCN
LRU
MRRI
MRRII
MRRMOD
MTD
NRTS
ORR
REFERENCE-NUMBER
REP-CYCLE-TIME
RIP
RISS-BUY
RMSS-LVL
RTD
RTLL
SCC
SMR

PRIMARY KEY

ALC
LCN
REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS PARENT

MAINTENANCE-PLANNING-FACTOR Can Be Overhauled At
REWORK-POINTS

IDENTIFYING RELATIONSHIP AS CHILD

PHYSICAL-LOCATION-WITHIN-SYSTEM Effects
MAINTENANCE-PLANNING-FACTOR

RATIO Z
FOREIGN KEY REFERENCE-NUMBER
FOREIGN KEY SCC
FOREIGN KEY ALC
FOREIGN KEY LCN

*** PACKAGING ***

GLOSSARY DESCRIPTION

Information pertinent to shipping and storing the item including size of item and its container and packing materials.

EXTENDED DATA

CD
CONTAINER-NSN
CT
CUSN-MATL
DOP
FSCM
HC
ICQ
INT-CONT
MTH-PRES
OPI
PCC
PK-CD
PRES-MATL
REFERENCE-NUMBER
SCC
SPEC-MKG
SPI-NO
SPI-REV
SUPP-PK-DATA
UC-LVL
UNIT-CONT
UNIT-SIZE
UNIT-WEIGHT
WRAP-MATL

PRIMARY KEY

PK-CD
REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE Has Packaging Requirements PACKAGING

FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** PHYSICAL-LOCATION-WITHIN-SYSTEM ***

GLOSSARY DESCRIPTION

The identifying information for an item's location within the hierarchy of an end item.

EXTENDED DATA

ALC
EC
LCN
QTY-ASSEMBLY
RDC
REF-DESIGNATION
REFERENCE-NUMBER
SCC
WUC-FGC

PRIMARY KEY

ALC
LCN
REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS PARENT

PHYSICAL-LOCATION-WITHIN-SYSTEM May be Found on
ILLUSTRATED-PARTS-BREAKDOWN
PHYSICAL-LOCATION-WITHIN-SYSTEM Effects
MAINTENANCE-PLANNING-FACTOR
RATIO Z

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE is Used in
PHYSICAL-LOCATION-WITHIN-SYSTEM
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER
WEAPON-SYSTEM Has Component Items in These
PHYSICAL-LOCATION-WITHIN-SYSTEM
FOREIGN KEY ALC
FOREIGN KEY LCN

*** POTENTIAL-SUPPLY-SOURCE ***

GLOSSARY DESCRIPTION

List of manufacturers that have the potential of producing the item.

EXTENDED DATA

CTIC
FSCM
REFERENCE-NUMBER
SCC

PRIMARY KEY

CTIC
REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE Has Fabrication Techniques to
Manufacture POTENTIAL-SUPPLY-SOURCE
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** PROCUREMENT-PRICE ***

GLOSSARY DESCRIPTION

The pricing structure of an item based on the quantity specified.

EXTENDED DATA

CPC
CSC-PRICE
FY
LOT-QTY
PUC
REFERENCE-NUMBER
SCC
TOTAL-QTY-RECOMMENDED
TUC
UM
UM-PRICE

PRIMARY KEY

CSC-PRICE
REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE Has Econ lot size Determined By
PROCUREMENT-PRICE
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** PROVISIONING-END-ITEM ***

GLOSSARY DESCRIPTION

The identifying information about an end item for a specific configuration.

EXTENDED DATA

ALC
LCN
PCCN
QTY-EI

PRIMARY KEY

ALC
LCN
PCCN

IDENTIFYING RELATIONSHIP AS PARENT

PROVISIONING-END-ITEM is Assembled From
PROVISIONING-LINE-ITEM
RATIO P

IDENTIFYING RELATIONSHIP AS CHILD

WEAPON-SYSTEM is Made up of PROVISIONING-END-ITEM
RATIO P
FOREIGN KEY ALC
FOREIGN KEY LCN

*** PROVISIONING-LINE-ITEM ***

GLOSSARY DESCRIPTION

Information about the relationship of a provisionable item with respect to its next higher assembly.

EXTENDED DATA

ALC
DRSC
IND-CD
LCN
NHA-IND
NHA-PLISN
PCCN
PLISN
PRIOR-ITEM-PLISN
REMARKS
SAME-AS-PLISN

PRIMARY KEY

ALC
LCN
PCCN
PLISN

IDENTIFYING RELATIONSHIP AS PARENT

PROVISIONING-LINE-ITEM Control Line Item Modifications
CHANGE-AUTHORITIES

IDENTIFYING RELATIONSHIP AS CHILD

PROVISIONING-END-ITEM is Assembled From
PROVISIONING-LINE-ITEM

RATIO P

LCN
FOREIGN KEY ALC
FOREIGN KEY PCCN

*** REWORK-POINTS ***

GLOSSARY DESCRIPTION

The physical location where an item can be repaired to working condition.

EXTENDED DATA

ALC
LCN
REFERENCE-NUMBER
REWORK-POINT
SCC

PRIMARY KEY

ALC
LCN
REFERENCE-NUMBER
REWORK-POINT
SCC

IDENTIFYING RELATIONSHIP AS CHILD

MAINTENANCE-PLANNING-FACTOR Can Be Overhauled At
REWORK-POINTS

FOREIGN KEY LCN
FOREIGN KEY ALC
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** STOCK-ISSUE-VALUATION ***

GLOSSARY DESCRIPTION

Information related to an item which basically relates the price of the provisionable item to its quantity per unit of measure.

EXTENDED DATA

QUP
REFERENCE-NUMBER
SCC
UI
UI-CONV-FACT
UI-PRICE
UM

PRIMARY KEY

REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE Has Current Issue Price of
STOCK-ISSUE-VALUATION
RATIO Z
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** STORAGE-AND-DISPOSAL ***

GLOSSARY DESCRIPTION

Data related to the disposal or retirement of a piece that has been replaced in a working system. This information may have bearing on the way it is stored or disposed.

EXTENDED DATA

ADP-EC
PMIC
PS-PC
REFERENCE-NUMBER
SCC
SL
SLAC
SMCC
SMIC

PRIMARY KEY

REFERENCE-NUMBER
SCC

IDENTIFYING RELATIONSHIP AS CHILD

ITEM-PRIMARY-REFERENCE Has Requirements For
STORAGE-AND-DISPOSAL
RATIO Z
FOREIGN KEY SCC
FOREIGN KEY REFERENCE-NUMBER

*** WEAPON-SYSTEM ***

GLOSSARY DESCRIPTION

Highest level tracked within the LSAR system. Includes the identifying data for a particular weapon system.

EXTENDED DATA

ALC
LCN

PRIMARY KEY

ALC
LCN

IDENTIFYING RELATIONSHIP AS PARENT

WEAPON-SYSTEM Includes Supporting ALLOWANCE-ITEM
WEAPON-SYSTEM May Have Functional CONFIGURATION
WEAPON-SYSTEM Has Component Items in These
PHYSICAL-LOCATION-WITHIN-SYSTEM
WEAPON-SYSTEM is Made up of PROVISIONING-END-ITEM
RATIO P

BUSINESS RULES REPORT

Every Additional-Vendor
always has an Item-Primary-Reference.

Every Affected-Configuration
always has a Change-Authorities.

Every Allowance-Item
always has a Weapon-System.

Every Basis-Of-Issue
always has a System Item-Primary-Reference.

Every Change-Authorities
always has a Provisioning-Line-Item.
Only Causes Modifications To zero, one or many
Affected-Configuration(s).

Every Configuration
always has a Weapon-System.

Every Illustrated-Parts-Breakdown
always has an Item-Primary-Reference.
always has a Physical-Location-Within-System.

Every Item-Primary-Reference
Form,Fit,Function Equivalent zero, one or many
Additional-Vendor(s).
Allowance Qty Defined for End Item zero, one or many
Basis-Of-Issue(s).
is Illustrated in Tech Manual zero, one or many
Illustrated-Parts-Breakdown(s).
Has Packaging Requirements zero, one or many Packaging(s).
is Used in zero, one or many
Physical-Location-Within-System(s).
Has Fabrication Techniques to Manufacture zero, one or many
Potential-Supply-Source(s).
Has Econ lot size Determined By zero, one or many
Procurement-Price(s).
Has Current Issue Price of zero or one
Stock-Issue-Valuation(s).
Has Requirements For zero or one Storage-And-Disposal(s).

Every Maintenance-Planning-Factor
always has a Physical-Location-Within-System.
Can Be Overhauled At zero, one or many Rework-Points(s).

Every Packaging
always has an Item-Primary-Reference.

Every Physical-Location-Within-System
always has an Item-Primary-Reference.
always has a Weapon-System.

May be Found on zero, one or many
Illustrated-Parts-Breakdown(s).
Effects zero or one Maintenance-Planning-Factor(s).

Every Potential-Supply-Source
always has an Item-Primary-Reference.

Every Procurement-Price
always has an Item-Primary-Reference.

Every Provisioning-End-Item
always has a Weapon-System.
is Assembled From one or more Provisioning-Line-Item(s).

Every Provisioning-Line-Item
always has a Provisioning-End-Item.
Control Line Item Modifications zero, one or many
Change-Authorities(s).

Every Rework-Points
always has a Maintenance-Planning-Factor.

Every Stock-Issue-Valuation
always has an Item-Primary-Reference.

Every Storage-And-Disposal
always has an Item-Primary-Reference.

Every Weapon-System
Includes Supporting zero, one or many Allowance-Item(s).
May Have Functional zero, one or many Configuration(s).
Has Component Items in These zero, one or many
Physical-Location-Within-System(s).
is Made up of one or more Provisioning-End-Item(s).

APPENDIX C - IRDS DDL statements

The IRDS is intended to be the repository for the comprehensive descriptions of data, data relationships, semantics and usage for a automated system, in this case the LSAR support item identification and end-item provisioning. Now that the enterprise model is complete, all information except usage is ready to be stored in an IRDS dictionary. An IRDS dictionary is loaded through its data definition language (DDL). Moving a a schema in or out of and IRDS dictionary would consist of a set of ADD statements that represent the total defininition of everything known about a system.

From the perspective of the IRDS, each atomic data unit is called an ENTITY. This is a different criteria than that of an entity in the E-R or IDEF1X data model sense. For the purpose of a data dictionary, even a data element may have properties about is such as valid codes, who controls its representation, and its last revision date. These properties about the ENTITY are called ATTRIBUTES. In examining the entity definition for Provisioning-Line-Item we notice that it is of type RECORD. The RECORD is the aggregation of all of the properties which describe an object or concept. After the entities are added to the schema the relationships between them must be established these statements are also included in this example.

```
ADD ENTITY      Provisioning-Line-Item
ENTITY-TYPE = RECORD
WITH ATTRIBUTES
  DESCRIPTION = "Information about the relationship of a
                provisionable item with respect to its next
                higher assembly" ;

ADD ENTITY      IND-CD
ENTITY-TYPE = ELEMENT
DESCRIPTIVE-NAME = Indenture-Code
WITH ATTRIBUTES
  DESCRIPTION = " ded-no 157 Indenture-Code      H10/08
                Illustrates a latest and descending 'family type'
                relationship of each line item to and within the system
                or end items and its discrete components (units).
                Assemblies and subassemblies and subsubassemblies. (The
                indenture code can be increased)." ;

ADD ENTITY      LCN
ENTITY-TYPE = ELEMENT
DESCRIPTIVE-NAME = Logistic-Support-Analysis-Control-Number
WITH ATTRIBUTES
  DESCRIPTION = "ded-no 197 Static for a physical
                decomposition (provisioning key element) Identifies
                where the component is in the system"
  DATA-TYPE = "static" ;
```

```

ADD ENTITY      NHA-Ind
ENTITY-TYPE = ELEMENT
DESCRIPTIVE-NAME = Next-Higher-Assembly-PLISN-Indicator
WITH ATTRIBUTES
  D E S C R I P T I O N      =      " d e d - n o      2 6 2
    Next-Higher-Assembly-PLISN-Indicator H10/13
    describes the NHA" ;

ADD ENTITY      PLISN
ENTITY-TYPE = ELEMENT
DESCRIPTIVE-NAME = Provisioning-List-Item-Sequence-Number
WITH ATTRIBUTES
  D E S C R I P T I O N      =      " d e d - n o      3 4 2
    Provisioning-List-Item-Sequence-Number
    H10/09 a key identifier for a line-item. PLISN and
    IND-CD together identifies same as an LCN." ;

ADD RELATIONSHIP
PROVISIONING-LINE-ITEMS RECORD-CONTAINS-ELEMENT PLISN ;

ADD RELATIONSHIP
PROVISIONING-LINE-ITEMS RECORD-CONTAINS-ELEMENT LCN ;

ADD RELATIONSHIP
PROVISIONING-LINE-ITEMS RECORD-CONTAINS-ELEMENT NHA-IND ;

ADD RELATIONSHIP
PROVISIONING-LINE-ITEMS RECORD-CONTAINS-ELEMENT IND-CD ;

ADD RELATIONSHIP
PLISN ELEMENT-DERIVED-FROM-RECORD PROVISIONING-LINE-ITEMS ;

ADD RELATIONSHIP
LCN ELEMENT-DERIVED-FROM-RECORD PROVISIONING-LINE-ITEMS ;

ADD RELATIONSHIP
NHA-IND ELEMENT-DERIVED-FROM-RECORD PROVISIONING-LINE-ITEMS ;

ADD RELATIONSHIP
IND-CD ELEMENT-DERIVED-FROM-RECORD PROVISIONING-LINE-ITEMS ;

```

APPENDIX D - IRDS Extensions

In addition to some of the basic schema of the IRDS extensions may be added such as new entity types or simply adding new attributes to existing entity definitions. In the support item identification and provisioning analysis extensions have been made to associate data types, valid data value ranges, and valid data value codes (domain properties) for the ELEMENT entity-type. This can be accomplished easily by the following set of IRDS statements or data definition language (DDL).

```
ADD META-ENTITY Data-Type
    META-ENTITY-TYPE = ATTRIBUTE-TYPE;
ADD META-RELATIONSHIP
    FROM ELEMENT TO Data-Type;
```

```
ADD META-ENTITY Codes
    META-ENTITY-TYPE = ATTRIBUTE-TYPE;
ADD META-RELATIONSHIP
    FROM ELEMENT TO Codes;
```

```
ADD META-ENTITY Range
    META-ENTITY-TYPE = ATTRIBUTE-TYPE;
ADD META-RELATIONSHIP
    FROM ELEMENT TO Range;
```

Once each of these 'META-ENTITIES' exist and have been linked by adding the 'META-RELATIONSHIP' they appear to the user as another attribute contained in the IRDS's basic attribute list and can be used when describing each of the data elements that make up the data model.

MEDIA

ICST RECOMMENDATIONS ON OPTICAL DISKS AND
INTERFACE REQUIREMENTS FOR PLANNED EDMICS PROCUREMENT
FINAL REPORT

ICST Recommendation on Optical Disk
and Interface Requirements for
Planned EDMICS Procurement
Final Report

May 29, 1987

(editorial corrections on
November 25, 1987)

ICST Recommendations on Optical Disks
and Interface Requirements for
Planned EDMICS Procurement
Final Report

Table of Contents

	<u>Page</u>
1.0 Background on Write-Once-Read-Many Times (WORM) Optical Digital Data Disk (OD ³) and Alternative Computer Storage Systems	1
1.1 Status of 5-1/4" (130 mm) WORM OD ³	3
1.2 Status of 12" (300 mm) WORM OD ³	5
1.3 Status of Other Computer Storage Media	5
2.0 Interchange Parameters to be Considered for OD ³ Computer Storage Systems	6
3.0 Care, Handling, and Life Expectancy of WORM OD ³	8
4.0 Background on Small Computer System Interface (SCSI)	8
5.0 Recommended Wording on SCSI and Computer Storage Systems for the Planned EDMICS Request for Proposals	9
5.1 SCSI-2 Physical Level Requirements	9
5.2 SCSI-2 Device Specific Interface Requirements	10
5.3 Removable Computer Storage Media Systems Requirements	10
6.0 Need for Continuing Support From ICST/NBS	11
7.0 Attachment - DDI Model for WORM OD ³	13

Figures

Figure 1	17
Figure 2	18

1.0 Background on Write-Once-Read-Many-Times (WORM) Optical Digital Data Disk (OD³) and Alternative Computer Storage Systems

The following information will address the 5 1/4" and 12" Write-Once-Read-Many-Times (WORM) optical storage computer systems that are of particular interest to the EDMICS program. Additionally, other potentially competitive computer storage systems that may be bid as a result of the issuance of an RFP will be documented. Within each specific WORM Optical Digital Data Disk (WORM OD³) computer storage system of interest to EDMICS, for the purpose of this report, we will explore the development background, available products, standardization efforts, recommended wording for an RFP, and discuss strategies for assuring ultimate standards.

In the report, references are made to documents submitted to ANSI Technical Committee X3B11 by specific companies and certain commercial equipment is identified. Such identification does not imply recommendation or endorsement by the National Bureau of Standards nor does it imply that the equipment identified is necessarily the best available for the intended purpose. These documents and the identified equipment are used in this report for reference only. The information was collected from technical and commercial literature and from personal contacts with standards committee members and industry representatives and to the best of our knowledge reflects the status of the issues that are included in this report.

The procurement of WORM OD³ storage systems, while the technology is in a development phase, may not be the most opportune time to acquire such systems if the use of the media is to be for the interchange of data utilizing the WORM OD³ in a multi-vendor environment. When a relatively new computer storage technology, such as WORM OD³, is being developed, there are many vendors who are interested in marketing a product that exhibits such tremendous opportunities for their respective companies. These divergent companies ultimately develop products that, while utilizing the identical storage medium, differ in physical size, recorded format, and user software for processing data on the medium. In other words, the implementation of the WORM OD³ product by vendor A is incompatible with the product of vendor B, thus, inhibiting and possibly prohibiting the interchange of information on WORM OD³ between the users of the products of vendor A and vendor B.

This situation is not unusual in the development stage of a new technology. The products have not had enough time to mature through applications in the marketplace to determine user acceptance. At first glance it might appear as though the easy answer for the incompatibility issue is to purchase products from one vendor's product line. If all users within an application have the same product, by definition, there can be no incompatibility. While this is true, there are shortcomings to this solution.

1. Government procurement regulations make sole source procurements difficult if not impossible.

2. If the sole source vendor goes out of business, there goes your support and supplier of hardware and software.
3. You may be required, in the future, to conduct business with someone who uses products that are incompatible with yours resulting in additional time and cost to accomplish the interchange.

The most opportune time for purchasing WORM OD³ systems would be when the technology has matured to the extent where stable implementations and products are available with substantial user and application support. The user and application support should be represented by the existence of standards that define and specify this medium so that vendors can supply and users can apply the technology to the maximum extent for the benefit of all concerned. This is especially true if the computer storage medium, WORM OD³ in this case, is to be used for the storage and interchange of engineering drawings within a multi-vendor application environment.

In the case of computer information processing standards within the United States, the voluntary standards are typically developed and approved by ANSI. Once approved by ANSI, and if deemed suitable for Federal Government use, they are approved as Federal Information Processing Standards (FIPS). The approval of such standards has tremendous impact upon the technology in that vendors have incentive to build products that are standard conforming for the user community. The users have expressed their desire for what they want in the product through the approval and support of a standard. Users can further express this desire and support of the standard by requiring vendors to provide products that are standard conforming when purchasing the products. In the case of the Federal government procurement this would be accomplished by citing the appropriate standard in the RFP.

The status of WORM OD³ technology, with respect to specifications to be inserted into an RFP, lies somewhere between development and maturity. As stated previously, there are no standards for WORM OD³. However, there are several proposals being considered for standardization by ANSI X3B11, the technical committee responsible for developing standards, both recorded and unrecorded, for WORM OD³. The following proposals have been submitted for WORM OD³ standardization and approved for development by ANSI X3B11.

1. Project #408-D 12 inch - Unrecorded optical media unit
2. Project #457-D 5.25 inch - Unrecorded optical media unit
3. Project #481-D 5.25 inch - Recorded optical media unit
4. Project #483-D 12 inch - Recorded optical media unit
5. Project #524-D all sizes - Label and file structure

1.1 Status of 5-1/4" (130 mm) WORM OD³

Regarding the 5 1/4" WORM OD³, there is presently a debate within ANSI X3B11 and industry on what type of format to adopt for the proposed draft standard with respect to the recorded format. There are two camps on this issue. One camp supports a sampled servo format and the other supports a continuous format.

Continuous Servo Tracking Format

Continuous servo tracking format (CSTF), also called the Composite Tracking Format, was the first method that was used in optical disks. Many manufacturers have practical experience with this method. CSTF implies that there is a continuous groove on the surface of the disk. The focusing servo information and the tracking signals are both drawn continuously from the groove. The data and servo signals are drawn at the same time.

The advantages of CSTF are:

- Insensitive to media defects
- Low overhead in servo and clock
- Good servo stability
- The groove on the disk can be copied with high accuracy
- Any coding format can be accepted
- Good experience in its implementation.

The disadvantages are:

- Interference between the servo and data channel.
- Expensive mastering for the groove.
- Hard to attain compatibility among different media and products.

Sampled Servo

The basic idea behind the Sampled Servo (SS) technique is to separate the various servo functions on an optical disk drive from each other and from the written data. In this system the servo and data signals are drawn at different times. No interference between data and servo signal occurs. For this to happen the servo technique should specify some extra servo areas (the so called synchro-servo zone) to be added to every sector.

The advantages of the SS method are:

- No interference between servo and data channels.
- A groove is not required.
- To master the servo areas is simple.
- It is easier to attain compatibility among different media and products.

The disadvantages are:

- Sensitive to defects on the media.
- High overhead in the form of servo and clock if high stability is desired.

Precise replication of wobbled pits in the entire disk surface might be difficult.

The aforementioned remarks are intended to give just a general idea of the difference between these two techniques. For more complete information on this issue see the following documents:

X3B11/86-190 1st Draft Proposed American national Standard 130 mm Optical Media Recorded Format Tracking and Servo Technique. Prepared by technical Committee X3B11, Recorded Format Ad-Hoc Group of Accredited Standards Committee X3.

X3B11/86-091 comments on Sampled Servo tracking and Continuous Servo Tracking (submitted by Hitachi).

X3B11/86-160 (submitted by Optimem).

It is necessary to note that these technologies are evolving and that it is inappropriate at the present time to give any judgement that one technique is better than the other. It is necessary to continue closely monitoring the evolution of these two technologies (and any other that would be submitted to the technical committees) until one or more of these methodologies becomes a standard.

An example of possible future alternative technologies in this area (physical format of the optical disk media) is in the recent announcement of IBM, the IBM PC System 2. The model 80 incorporates a 200 Mbyte, singled sided optical disk drive that supports media whose physical format is not compatible with either camp (is not compatible with the continuous format or the sampled servo proposal in X3B11).

However, this is not expected to become a third proposal in the immediate future. An IBM member of X3B11 stated recently that IBM has no plans for a technical submission to X3B11 in this area. This will probably not happen until the product is in the market which will not occur until 6 to 9 months from now.

With respect to the two proposed formats, the two groups emphasize that current technical debates will not deter standardization efforts. Both agree some sort of standard is necessary for the growth of the optical market. But the fact that the two camps are so far apart does create problems. Going with either proposal means that companies will have to revise next-generation products to match the standard, whenever it is approved. Users will also be negatively impacted by the current lack of standards.

Some of the companies in support of the sample servo format approach are:

1. Alcatel-Thomson Gigadisk (ATG)
2. Laser Magnetic Storage (LMS)
3. Sony Corporation

Some of the companies in support of the continuous servo format approach are:

1. Maxtor
2. Optotech
3. Hitachi

At the most recent meeting of ANSI X3B11, March 11-12, 1987 in Portland, Oregon, the committee approved a motion that "X3B11 resolves to take the necessary steps to have a project approved for each of the formats being considered under an appropriate title such as:

Unrecorded Optical Media Unit for Digital Information Interchange
(130mm, Sampled Servo Method)

Unrecorded Optical Media Unit for Digital Information Interchange
(130mm, Continuous Servo Method)".

1.2 Status of 12" (300 mm) WORM OD³

The level of standardization activity on 12" WORM OD³ is considerably less than the effort on 5 1/4" WORM OD³. The ANSI X3B11 Technical Committee has approved two projects on the 12" WORM OD³. Both projects are over two years old and very little work has taken place. The only activity worth noting is that before the X3B11 January 1987 meeting, NBS was asked if the Government still had interest in the 12" WORM OD³. NBS responded in the affirmative. Other members of X3B11 also expressed renewed interest in the 12" WORM OD³. In other words, X3B11 shows some intent to reactivate these projects. To date, no new documents have been submitted.

1.3 Status of Other Computer Storage Media

As a result of the issuance of an RFP for WORM OD³ computer storage, products other than 5 1/4" WORM OD³ and 12" WORM OD³ may be bid. Within the X3B11 community the following projects have been approved.

- | | |
|-------------------|---|
| 1. Project #407-D | 8" Unrecorded optical media unit |
| 2. Project #409-D | 4.72" Unrecorded optical media unit |
| 3. Project #456-D | 14" Unrecorded optical media unit |
| 4. Project #480-D | 4.72" Recorded optical media unit |
| 5. Project #482-D | 8" Recorded optical media unit |
| 6. Project #484-D | 14" Recorded optical media unit |
| 7. Project #581-D | 3.5" Unrecorded reversible optical media unit |

The 8" WORM OD³ product is a popular size in Japan. The Panafax Company recently exhibited a drive using the 8" WORM OD³ at the Association for Information and Image Management (AIIM) Exhibition in New York in April 1987. The product consisted of a Matsushita drive using a Matsushita 8" optical disk. The Eastman Kodak Company recently announced a product that

utilizes a 14" optical disk. The system, "Kodak Optical Disk System 6800", offers 6.8 gigabytes of storage on a single optical disk cartridge and up to 1020 gigabytes in an automated library. It is expected to be available in the Fall of 1987.

At the AIIM Exhibition company representatives were questioned about the existence of second sources for optical media and drives. All responded that a second source was not available at this time. Some company representatives suggested that in order to assure media availability, the best solution is to stock enough media and have at least two subsystems with media of different manufacturers and computer software to copy from one media to the other in case one media would become unavailable.

It is worthwhile to note that there are efforts in X3B11 to establish a method of assuring data interchange using one drive and different media manufacturers. The effort suggests the inclusion of a control track on the media that would provide identification of the media type. This information would allow a drive to recognize which type of media is being used and would produce the necessary adjustments to utilize the media. At this time, however, the effort is concentrated on the 5 1/4" WORM OD³ only.

There are magnetic tape and magnetic tape cartridge computer storage systems that have the potential of being competitive with the WORM OD³ storage systems in some applications. The IBM 3480 Tape Cartridge system is typically used in a backup and archival application and for a particular application may be competitive with WORM OD³ on a cost per byte basis. Storage Technology Corporation's 4400 Automated Cartridge System has attributes similar to the IBM 3480 system.

This discussion of WORM OD³ storage systems, other than 5 1/4" and 12" systems, and the discussion of magnetic tape cartridge storage systems is not presented as an all inclusive and exhaustive study. The purpose is to illustrate that there are other computer storage systems that may be competitive with the 5 1/4" and 12" WORM OD³ technologies and the EDMICS program should anticipate that one or all of these alternatives may be bid as a result of the issuance of an RFP. The EDMICS program should also be prepared to evaluate these technologies as they relate to the programmatic requirements.

2.0 Interchange Parameters to be Considered for WORM OD³ Computer Storage Systems

In order to interchange specialized digital information on removable computer storage media, several levels of interchange parameters must be specified. ICST has recently introduced to national (X3) and international standards organizations a proposed Reference Model for Digital Data Interchange (DDI) via removable computer storage media.

Figure 1 outlines the structure (levels 1-4) and conveys the hierarchical nature of the proposed DDI Reference Model. Level 1 specifies the interchange requirements for unrecorded or unformatted media. Level 2

specifies the interchange requirements of the recorded or formatted media. Level 3 specifies the interchange requirements for the volume identification labels, file directories, and file structures of the recorded media. These first three levels of standardization are the minimum number sufficient to ensure data interchange between information processing systems via removable computer storage media. Level 4 in this reference model is required in order to accomplish specialized tasks, such as interchanging text on flexible disk cartridges or interchanging images on optical disks.

The boundaries between these levels can vary somewhat, depending on the characteristics of a particular type of removable computer storage media technology. However, these boundaries tend to occur as a natural consequence of the component parts (hardware and software) which comprise an information processing system. Figure 2 illustrates that each interchange level in the proposed DDI Reference Model impacts the design of one or more of the following component parts of an information processing system: a peripheral storage subsystem (i.e., medium, drive, and controller), an operating system, or a specialized software function (e.g. word processor, CAD-CAM, etc.). Because of the resulting matrix of interchange levels and impact on information processing system design, it is highly unlikely that a single group of technical experts will possess the breadth and depth of knowledge necessary to develop interchange standards for levels 1 through 4 of the proposed DDI Reference Model. To date, there have always been more than one group of experts necessary in order to develop standards at levels 1 through 4.

Attached to the end of this report is a detailed set of interchange parameters (levels 1-3) which need to be specified for data interchange via WORM OD³.

The attachment requests bidders to specify any additional parameter requirements they have for data interchange.

The parameter list in the attachment is divided into different categories. Subdivisions in these different categories are a guideline. Different bidders could have other ways of specifying similar parameters, and in each particular case it will be required to study the different tests and parameter descriptions, values and tolerances, engineering drawings, etc.

Parameters specified for levels 1 and 2 are according to the parameters specified in X3B11 draft proposals such as documents X3B11/86-151R3 (03-12-87) for unrecorded optical media unit for Digital Information Interchange (130 mm Optical Disk Cartridge) and documents X3B11/86-190R3 (05-01-87) Proposed American National Standard, 130 mm Optical Media Part 4, Recorded Format, Tracking and Servo Technique.

The parameters list specified by bidders for other disk sizes such as 8", 12", and 14" will vary according to specific requirements for each particular size. The following key indicates how the parameters are specified in the attachment:

- A Number and tolerances or nominal or maximum, minimum value
- B Descriptions and/or Figures
- C Detailed Description of test procedures
- D Engineering Drawings
- E Byte Count-Relative Position

3.0 Care, Handling, and Life Expectancy of WORM OD³

At the present time there are no standards or recommended practices regarding the care, handling, and life expectancy of WORM OD³. Vendors who bid products as a result of the EDMICS RFP should be required to address these issues to the best of their ability. Without knowing the exact application and requirements of the WORM OD³ within the EDMICS program, it is difficult to state how critical these issues may become. However, it is safe to state that one does not want to ascertain the useful life of data on the WORM OD³ by calculating the time between the creation of the data and the occasion of loading the WORM OD³, containing critical data, and discovering that it is unreadable.

4.0 Background on Small Computer Systems Interface

Many commercially available optical disk, magnetic disk, magnetic tape, and scanner products currently use the Small Computer System Interface (SCSI) standard. American National Standard X3.131-1986 defines the original version of this interface, however, since its publication Task Group X3T9.2 is further developing this now widely used standard, and the new version is informally referred to as "SCSI-2".

In SCSI-2 X3T9.2 has considerably refined the Command Set for Write-Once Read-Multiple Devices (optical disk drives), has added a Command Set for Scanner Devices and intends to add a Command Set for Changer Devices (for removable media storage devices). By and large, the industry is tracking developments in this committee and currently ongoing new product development reflects the "SCSI-2" draft rather than X3.131-1986. In most respects SCSI-2 compatible products will be compatible with X3.131-1986, however products claiming conformance to X3.131-1986 may not meet many of the requirements of SCSI-2. Although SCSI-2 is not finalized, it is anticipated that it will be finalized before volume purchases occur under this solicitation and that the necessary computer and storage equipment conforming to SCSI-2 will be widely available at that time. The current goal for the forwarding of the proposed SCSI-2 standard from Task Group X3T9.2 is January 1988.

SCSI-2 is defined to be the "Working draft American National Standard for information systems - enhanced SMALL COMPUTER SYSTEM INTERFACE (SCSI-2)," document X3T9.2-109 Rev. 1 dated April 15, 1987 or more current revision as appropriate. New revisions shall be deemed to supersede previous revisions upon their acceptance by X3T9.2 as the working draft and the working drafts will in turn be superseded by the published American National Standard for enhanced Small Computer System Interface (SCSI-2), X3.131-198X. In effect bidders shall commit to track the developing SCSI-2 standard and ultimately

deliver computer and storage equipment which conforms to the final published ANSI standard. This is considered appropriate because the computer industry is itself tracking the SCSI-2 standard with its products.

Current copies of SCSI-2 drafts are available as "X3.131-198X" from:

Global Engineering Documents
2625 Hickory Street
Santa Anna, CA 92707
800-854-7179
714-540-9870

5.0 Recommended Wording on SCSI and Computer Storage Systems for the Planned EDMICS Request for Proposals

All magnetic tape drives, optical disk drives, and scanners shall be connected to host computers via the physical and logical interface specified in SCSI-2, sections 4 and 5. It is desirable, but not required, that magnetic disk drives also be connected to host computers via the SCSI-2 physical and logical interface.

All products bid as a result of the RFP shall be required to conform to SCSI-2. However, the first delivery of equipment may conform to SCSI with the stipulation that the initial delivery be brought into conformance with SCSI-2 within one year. All products delivered subsequent to the initial delivery shall conform to SCSI-2.

5.1 SCSI-2 Physical Level Requirements

5.1.1 SCSI Connectors.

To ensure flexible configurability of the EDMICS hardware, all separate stand alone enclosures connected by shielded SCSI-2 cables shall provide two external connectors of the type identified as "Alternative 2" in Appendix D of ANSI X3.131-1986 (this connector is known generically as a 50 position miniature ribbon connector) and shall pass the SCSI-2 bus through the enclosure allowing daisy-chaining of enclosures, except that physically small desk top computers may provide only a single external SCSI-2 connector.

5.1.2 SCSI-2 Terminators.

Wherever enclosures are connected by shielded cables, external terminators which plug into external connectors, rather than internal terminators, shall be provided. Physically small desk-top computers may provide only a single SCSI connector and utilize internal terminators. This requires that such computers be cabled only on one end of a bus, but may be restricted by connector size constraints. All other computers shall provide two connectors and external terminators.

5.1.3 Setting SCSI Bus Addresses.

It is desirable, but not required that it be possible to set the bus address of all SCSI devices by some externally accessible means, such as

thumb wheel switches, rather than by opening the enclosure and setting DIP switches, straps and the like on printed circuit cards.

5.1.4 SCSI-2 Electrical Signals.

SCSI-2 offers two (incompatible) electrical alternatives. Single-Ended SCSI-2 buses are limited to a maximum length of 6m. Differential SCSI-2 buses have a maximum length of 25m, and considerably greater common-mode noise rejection. The circuitry for single-ended SCSI-2 is inherently less expensive. Work stations or small clusters of equipment which may be contained in a single rack or enclosure or upon a single desk top and which are powered from a single wall outlet may utilize either electrical option. All other equipment shall utilize the differential option for any SCSI-2 bus requiring external cables.

5.2 SCSI-2 Device Specific Interface Requirements

Requirements specific to each specific class of peripheral device are specified below:

5.2.1 Magnetic Disk Drives.

It is desirable but not required that magnetic disk drives employ the SCSI-2 interface and execute the Command Set for Direct-Access Devices specified in Section 8 of SCSI-2. Where magnetic disk drives do not employ the SCSI-2 interface they shall employ one of the approved ANSI standard disk drive interfaces such as:

- (1) the Storage Module Drive (SMD) interface, ANSI X3.91M-1987,
- (2) the Intelligent Peripheral Interface (IPI) Physical Level, X3.129-1986, plus either the IPI Device Generic Command Set for Magnetic and Optical Disk, X3.147-1987 or the IPI Device Specific Command Set for Device Specific Magnetic disk, X3.130-1986, or
- (3) the proposed Enhanced Small Drive Interface (ESDI) standard now under development in Task Group X3T9.3 (ESDI conforming drives are commercially available).

5.2.2 Magnetic Tape Devices.

All magnetic tape devices shall employ the SCSI-2 interface and execute the command set for Sequential Access Devices specified in Section 9 of SCSI-2.

5.2.3 Scanners.

All page scanner devices shall employ the SCSI-2 interface and execute the command set for Scanner devices specified in section 14 of SCSI-2.

5.3 Removable Computer Storage Media Systems Requirements

The following is the recommended wording for the purchase of the WORM OD³ computer storage systems.

5.3.1 5.25 inch WORM.

All 5.25 inch Write-Once-Read-Many-Times (WORM) peripheral storage subsystems shall be identified as either continuous servo or sample servo (in accordance with X3B11 documents X3B11/86-103R1, 151-R2, and 190-R1, or latest editions) or other type of WORM subsystems. Second sources for compatible media and drives shall be identified. Each 5.25 inch WORM subsystem bid shall include a complete set of level 1 through 3 (inclusive) data interchange requirements as outlined in the attached DDI Reference Model for WORM optical disk technology. The government retains the rights to this information so that, at the government's option, this information can be provided by the government to accredited standards organizations in order to promulgate a set of standards for ensuring data interchange via removable WORM technology.

5.3.2 Mass Storage.

All mass storage subsystems which use removable media (e.g., 12 inch WORM or 14 inch WORM, or magnetic tape) shall identify second sources of media. Each mass storage subsystem bid which uses WORM media shall include a complete set of level 1 through 3 (inclusive) data interchange requirements as outlined in the attached DDI Reference Model for WORM optical disk technology. The government retains the rights to this information so that, at the government's option, this information can be provided by the government to accredited standards organizations in order to promulgate a set of standards for ensuring data interchange via removable WORM technology.

6.0 Need for Continuing Support From ICST/NBS

While this report reflects as accurately as possible the current status of WORM OD³ technology for the purpose of preparing an RFP for the procurement of WORM OD³ storage systems, one invariable conclusion is the need for additional utilization, research, and standardization efforts regarding WORM OD³. The WORM OD³ storage systems represent great potential for vendors and users in the computer industry, but, thus far, the technology has not achieved widespread use due to the aforementioned lack of standards, knowledge on life expectancy of the media, and acceptance by a large user base. The award of a contract as a result of any RFP issued by the EDMICS program may well serve as the catalyst within the WORM OD³ industry (both users and vendors) and result in a defacto standard. This would be a significant milestone in the evolution of the WORM OD³ technology.

However, care must be taken that the previously mentioned areas of concern be addressed and resolved within the industry. These areas include:

1. Coalescence and agreement on a recording format for WORM OD³ (sampled, continuous, etc.) through the voluntary standards process of ANSI, ISO, ECMA that ultimately results in a FIPS for the Federal Government.

2. Coalescence and agreement on standards at all levels of the DDI reference model for WORM OD³ technology.
3. Research into the care, handling, and life expectancy of WORM OD³ to enhance widespread use and promote credibility of the media for long term use in applications.

The ICST is uniquely qualified to assist the EDMICS program in achieving reasonable solutions to these issues through its representation on the relevant voluntary standards committees; research within the ICST laboratories on WORM OD³ media, software, and hardware; and the publication and promotion of the results in the computer industry.

It is expected that ICST members will periodically support other government agency efforts on OD³ technology such as in this case the EDMICS project. The ability of ICST members to successfully react to those requests and have a weighted opinion on these issues is based on the possibility of them to keep track of this evolving technology.

The following are some of the tasks that ICST considers to be necessary in order to successfully perform as an advisory group in OD³ technology.

- a) Literature search, investigation and study of research papers on OD³ technology.
- b) Study of standards committee documents such as X3B11 documents, technical submissions, detailed study of testing procedures and draft proposals, in conjunction with frequent contacts with standards committee members and industry representatives.
- c) Participation in X3B11 committee meetings such as ad-hoc group meetings and plenary sessions, and other related technical meetings.

The present ICST budget has limited resources to fulfill these requirements. If other agencies expect to receive quality advice and support on these issues, they will have to fund these efforts in order to allow ICST members to keep up with this evolving technology.

7.0 Attachment - DDI Model for WORM OD³

The attached parameter list is divided into different parameter categories (A, B, C, D, and E). Subdivisions in these different categories should be considered as a guideline. Bidders may have other ways of specifying similar parameters.

Bidders can document these parameters in different categories. However, all of the following information shall be included. Bidders are required to specify any additional parameter requirements they have for data interchange.

Parameters specified for levels 1 and 2 are according to the parameters specified in X3B11 draft proposals, such as document X3B11/86-151R3 (03-12-87) for Unrecorded Optical Media Unit for Digital Information Interchange (130 mm Optical Disk Cartridge) and document X3B11/86-190R3 (05-01-87) Proposed American National Standard, 130 mm Optical Media Part 4, Recorded Format, Tracking and Servo Technique.

The following key indicates how the parameters are to be specified:

- A Number and tolerances, or nominal, or maximum, minimum value
- B Descriptions and/or Figures
- C Detailed Description of Test Procedures
- D Engineering Drawings
- E Byte Count - Relative Position

Level 1 - Media:

(a) Mechanical characteristics of the disk

Dimensions :

Outer diameter	A
Center hole	A
Concentricity	A

Clamping zone:

Outer diameter	A
Inner diameter	A

Disk total thickness without hub	A, B
Mass	A
Moment of inertia	A
Imbalance	A
Dynamic axial runout	A, B
Acceleration of axial runout	A
Dynamic radial runout	A
Acceleration of radial runout	A

Static deflection	A, B
Clamping technique	A, B
Clamping force	A
Hub dimensions	A, D
Location of the recording layer	A

(b) Mechanical Characteristics of the case

Mass	A
Shutter opening force	A
Nominal dimensions :	A
Length	A
Width	A
Thickness	A

(c) Optical characteristics of the disk

Optical characteristics of the protective layer :

Index of refraction	A
Thickness	A, B, C
Birefringence	A, B, C
Tilt	A, B

Optical characteristics of the recording layer :

Baseline reflectivity	A, B
Uniformity of base reflectivity	A, B

(d) Read/write characteristics

Optical conditions: A, B

Wavelength	
Filling of the lens aperture	
Beam power variance of the wavefront	

Reflectivity characteristics:

Unrecorded user area : A, B

On track	
At midpoint between two adjacent unrecorded tracks	
Control track	
Unrecorded user area without tracks	
Unrecorded user area with tracks	
Recorded user area (post-recorded)	
Recorded non-user area	

Recorded user area: A, B

Laser writing (Post recorded)
Low to high reflectivity
High to low reflectivity

Properties of the pre-recorded signals: A, B

Polarity
Reflectivity
Reflectivity amplitude change
Carrier-to-noise ratio and C
Cross-talk level and C

Read characteristics : A, B

Laser read power:

Hole open - High range
Hole close - Low range
Duty cycle

Write characteristics : A, B

Write power range

(e) Testing Procedures C

Level 2 - Physical Format on the media:

(a) Format

Modulation method B
Type of track B

Track format: B

Track provision
Direction of rotation
Track pitch and A
Recording location
Track format description
Total bytes/track and A
Servo bytes/track and A

Sector format:

Sector size	A
Total bytes/sector	A
User bytes/sector	A
Sectors/track	A
Description of the prerecorded preformatted information	B
(b) EDAC	B
Read characteristics	
Recording zone :	A, B
Lead-in area	
User recordable area	
Guard band	
Sector field functions	B
(c) Modulation Schemes	B

Level 3 - Logical Format on Media:

(a) Logical Volume Labels

Volume Identifier	E
User Identifier	E
Accessibility	E
Accounting Information	E
Directory	E

(b) Logical File Structure

File Identifier	E
User Identifier	E
Accessibility	E
Record Structure	E
Directory	E
Data Coding (e.g., ASCII)	E

**DIGITAL DATA INTERCHANGE (DDI) REFERENCE MODEL
FOR REMOVABLE COMPUTER STORAGE MEDIA**

- LEVEL 4 SPECIAL APPLICATION
 ON MEDIA**
 - control codes for word processing
 - data compression schemes for
 raster-scanning documents
 - etc.

- LEVEL 3 LOGICAL FORMAT FOR MEDIA**
 - logical volume labels
 - logical file structures
 - etc.

- LEVEL 2 PHYSICAL FORMAT ON MEDIA**
 - track locations
 - modulation schemes
 - data correction techniques
 - data code (e.g., ASCII)
 - etc

- LEVEL 1 MEDIA**
 - optical properties
 - magnetic properties
 - mechanical properties
 - dimensional properties
 - etc.

Figure 1

DIGITAL DATA INTERCHANGE (DDI) REFERENCE MODEL
FOR REMOVABLE COMPUTER STORAGE MEDIA

LEVEL OF DIGITAL DATA INTERCHANGE STANDARD	LEVEL OF IMPACT ON INFORMATION SYSTEM DESIGN			
	MEDIA	DRIVE CONTROLLER	OPERATING SYSTEM	SPECIALIZED SYSTEM
SPECIAL APPLICATION ON MEDIA				
control codes for word processing	--	--	?	x
data compression schemes for raster scanning	--	--	?	x
LOGICAL FORMAT FOR MEDIA				
logical volume labels	--	?	x	?
logical file structures	--	?	x	?
PHYSICAL FORMAT ON MEDIA				
track locations	x	x	?	--
modulation schemes	x	x	?	--
data correction techniques	x	?	?	--
data code (e.g. ASCII)	--	?	x	--
MEDIA				
optical properties	x	x	?	--
magnetic properties	x	x	x	--
mechanical properties	x	x	x	--
dimensional properties	x	x	?	--

KEY: x = definite impact; ? = possible impact; -- = no impact

Figure 2

RASTER COMPRESSION

REPORT ON RASTER GRAPHICS

TILED RASTER INTERCHANGE FORMAT, TRIF VERSION 1.0, REVISION 1.2

**REPORT ON
RASTER GRAPHICS**

September 30, 1987

RASTER GRAPHICS

INTRODUCTION

The purpose of this report is to describe the activities that have occurred in the area of raster graphics during FY 1987. There were two primary areas of concentration during the year: (1) Conformance Testing of CCITT Group 4 Compression, and (2) Tiling.

CONFORMANCE TESTING OF CCITT GROUP 4 COMPRESSION

At the request of the CALS Policy Office, NBS has established an interim testing procedure to evaluate conformance to the CCITT T.6 Group 4 compression/decompression algorithm. NBS installed software that will perform the evaluation and agreed to conduct the CALS demonstration conformance testing during the period June through September 1987.

The test procedures used by NBS during this demonstration period are illustrated in attachment 1. The "Testing Authority" is the site responsible for performing the evaluation. The "Remote Implementation Under Test" may be any DoD data repository or DoD contractor site. A description of the testing procedures follows.

The "Remote Implementation Under Test" submits a request to the "Testing Authority" who returns the hard copy test image(s) to be used for the test along with specifications for recording the test results on magnetic tape.

The "Remote Implementation Under Test" scans the hard copy test image(s), compresses the data (supposedly using T.6 Group 4 compression algorithm), records the compressed image on magnetic tape, and sends the test data tape to the "Testing Authority."

The "Testing Authority" reads the test data tape, decompresses the data, displays the resulting image, and visually compares the results with the original hard copy image(s).

The "Testing Authority" notifies the "Remote Implementation Under Test" of the testing results. A "pass" verdict by the "Testing Authority" signifies conformance.

During the demonstration period, images compressed using the Group 4 compression algorithm were sent to NBS and NBS successfully decompressed the data images and printed the

results. Two of the images were received from the Planning Research Corporation who are under contract to the Patient Trade Office (PTO) for raster graphics effort. Attachment 2 is a PTO Image System Test Target and attachment 3 is one of the PTO patients. Similarly, NBS also processed an image from Delta Information Systems, see attachment 4. The reduced image sizes in attachments 2 and 3 illustrates the difference of density resolutions between the bit-mag image and the capabilities of the printer.

TILING

A tiling scheme has been developed which supports the interchange of large format engineering drawings in raster format. This section of the report describes the background, methodology and major features of the tiling scheme.

Background

Interest in an industry-supported tiling scheme was first expressed at a meeting with DoD on April 15, 1987. As a result of that meeting, a task group composed of industry representatives including system integrators, peripheral manufacturers, and users as well as government representatives, assembled in an open forum to exchange views on tiling. This ad-hoc group, Tiling Task Group (TTG), concluded that development of an interchange format using tiling was desirable. Subsequently, a number of meetings and reviews were held by the TTG in order to develop a standard interchange format. The group has completed a draft Tiled Raster Interchange Format (TRIF), September 30, 1987 (see Appendix A) and will be completing a letter ballot vote in early FY 1988.

Methodology

The TTG developed the interchange format standard using existing and emerging standards as a basis wherever possible, especially the ISO 8613 series of documents on Office Document Architecture (ODA) and Interchange Format. This strategy assured that efforts were within the mainstream of raster imaging standards and promoted interoperability with other raster graphics formats utilized in related office document architecture standards.

Major Features

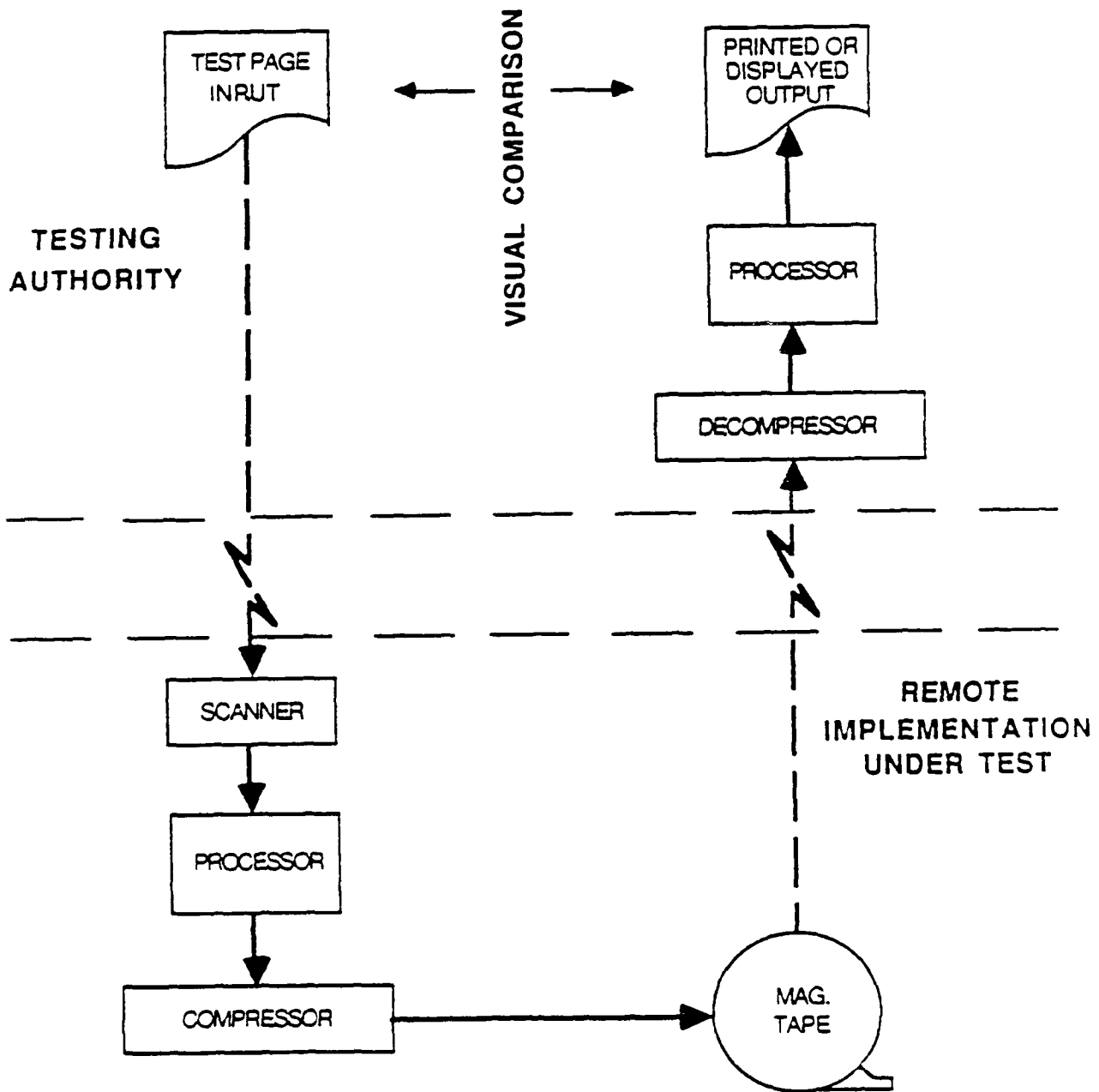
In an effort to keep the interchange format as simple as possible, a number of parameters such as tile size were fixed. It was decided that the only capability to remain negotiable between interchanging parties was resolution.

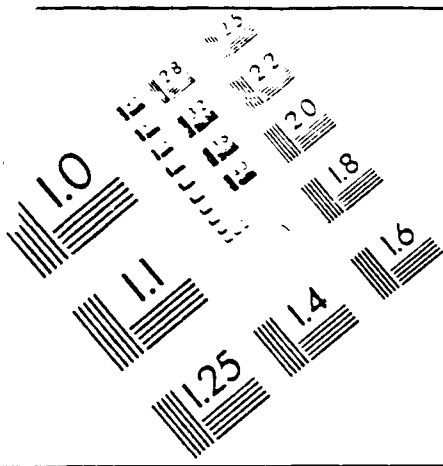
The interchange format deals only with bi-tonal (black and white) data. A tile is a block in a page in which all blocks have the same dimensions. All tiles are square and a single size is specified, 512 by 512 picture elements (pels). Any given tile is only to be encoded as CCITT Group 4 compressed data or as ISO 8613 bit-map data.

Media and database issues were considered to be outside the realm of this standard and were specifically excluded from consideration.

Appendix E of the TRIF document describes in more detail the goals and intent of the TTG.

COMPRESSION ALGORITHM CONFORMANCE TESTING



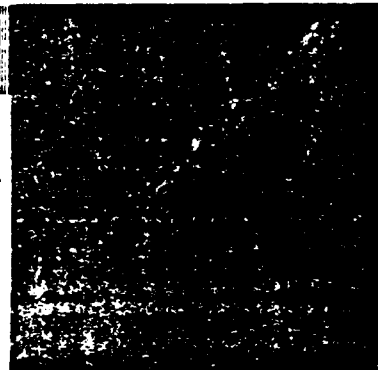
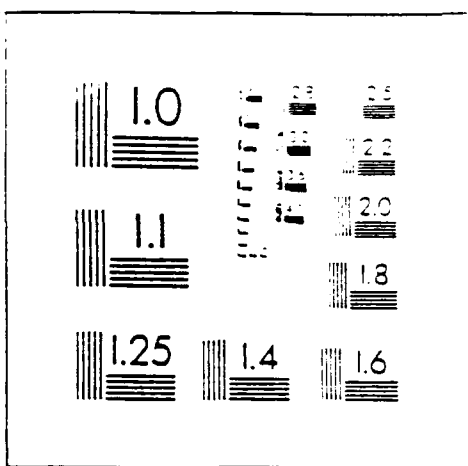


10 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 1234567890

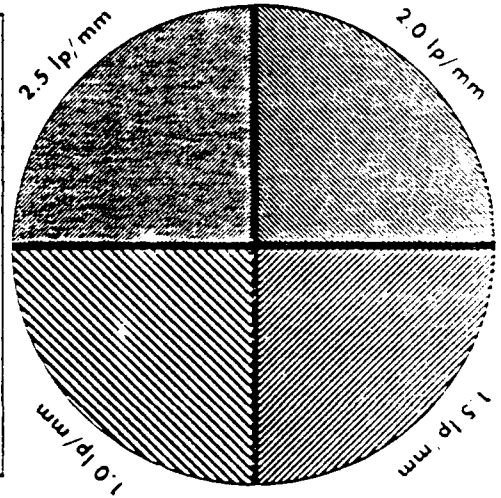
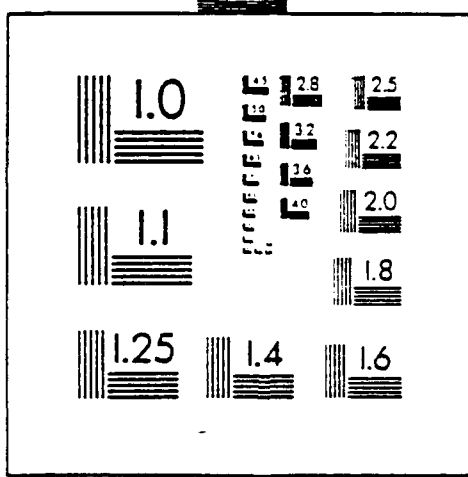
8 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890

6 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890

4 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890



2.5 lp/mm

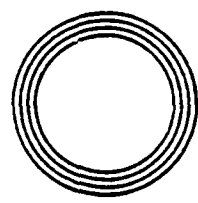


10 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 1234567890

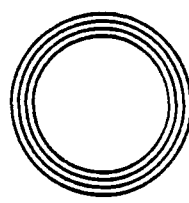
8 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890

6 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890

4 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890



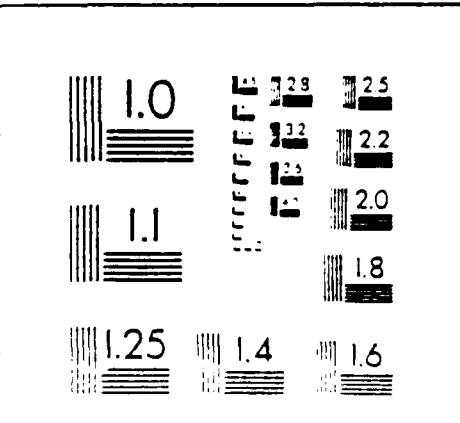
1.0 lp/mm



1.0 lp/mm



IMAGE SYSTEM
 TEST TARGET
 U.S. DEPARTMENT OF COMMERCE
 PATENT AND TRADEMARK OFFICE

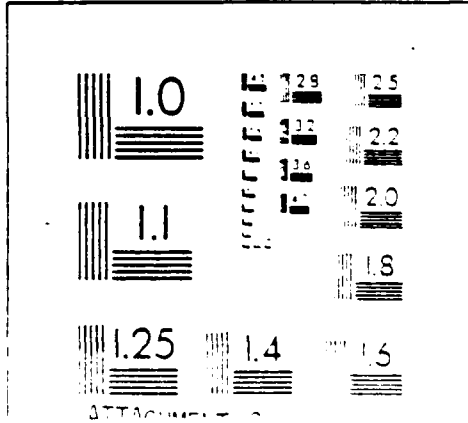


10 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 1234567890

8 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890

6 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890

4 PT. ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz1234567890



Nov. 24, 1953

S. BERTRAM
SWEEP CIRCUIT

2,660,691

Filed Aug. 17, 1944

3 Sheets-Sheet 1

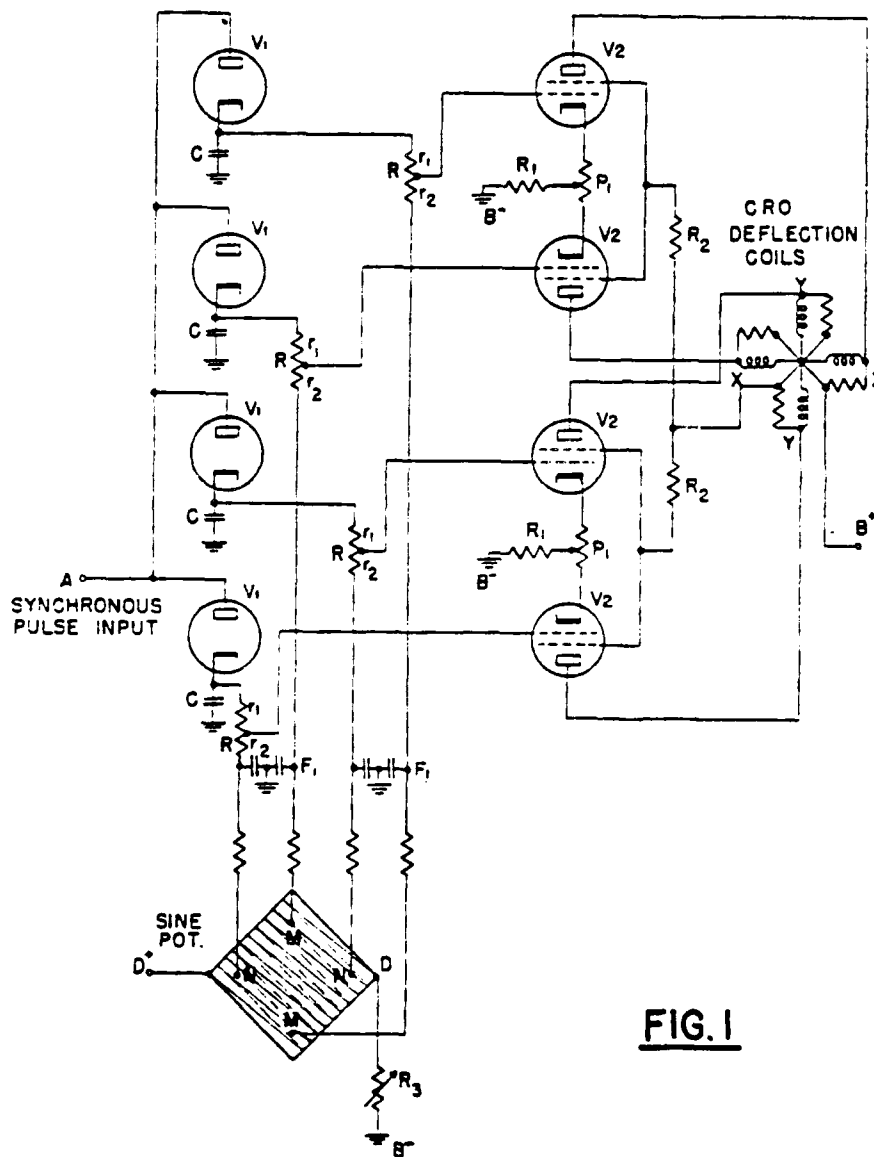


FIG. 1

INVENTOR.
SIDNEY BERTRAM
BY *[Signature]* ATTACHMENT 3
ATTORNEY



THE SLERENE COMPANY LIMITED

SAPORS LANE . BOOLE . DORSET . BH 25 8ER

TELEPHONE BOOLE (045 13) 51617 . TELEEX 123456

Our Ref. 350/PJC/EAC

18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photo cell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photo cell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Registered in England No. 2014
Registered Office: 30 Victoria Lane, Ilford, Essex.

ATTACHMENT 4

Tiled Raster Interchange Format
TRIF Version 1.0

Produced by the Tiling Task Group
September 30, 1987
Arlington, Virginia

Sponsored by the National Bureau of Standards

CONTENTS

- 1 *General*
 - 1.1 Scope
 - 1.2 Introduction
 - 1.3 General Characteristics of the Interchange Format
- 2 *Functions for the structure and interchange of text*
 - 2.1 Document profile
 - 2.2 Specific layout structure
 - 2.3 Document interchange
 - 2.4 Text positioning
 - 2.5 Attributes
- 3 *Functions for the interchange of text in processable form*
- 4 *Specification of protocol elements*
 - 4.1 Protocol elements
 - 4.2 Document profile descriptor
 - 4.3 Presentation capabilities descriptor
 - 4.4 Layout descriptor
 - 4.5 Text units
- 5 *Application rules*
 - 5.1 Interchange formats
 - 5.2 Order of transmission
 - 5.3 Interchange format TRF.0
 - 5.4 Values for negotiable facilities for TRF.0
- Annex A — Terms and definitions
- Annex B — Summary of presentation transfer syntax
- Annex C — Summary of data element identifier assignment
- Annex D — Coding Examples
- Annex E — Tiling Task Group goals and intent
- Annex F — Application of the Tile Index

- 1 *General*
- 1.1 *Scope*
- 1.1.1 This recommendation defines technical requirements unique to the tiled facsimile mode of document interchange.
- 1.1.2 The current understanding of the relationship between this recommendation and other OSI protocols is illustrated in Figure 1/TTG-87/42.

	Document interchange protocol	Rec. TTG/87-42
OSI session service	OSI session protocol	Rec. X.215 Rec. X.225
OSI transport service	OSI transport protocol	Rec. X.214 Rec. X.224
OSI network service		Rec. X.213

Figure 1/TTG.87-42

Relation between Recommendation TTG/87-42 and OSI protocols

(This figure is intended to be a guideline for the further work and requires further study)

- 1.1.3 The following CCITT Recommendations may also relate to the tiled facsimile mode of operation:
- T.5: General aspects of Group 4 facsimile apparatus;
 - T.6: Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus;
 - T.62: Control procedures for the Teletex and Group 4 facsimile services;
 - T.70: Network-independent basic transport service for telematic services;
 - T.73: Document interchange protocol for telematic services;
 - Series I Recommendations as applicable.
 - Series T.400 Recommendations as applicable.
- 1.1.4 The following ISO standards may also apply to the tiled facsimile mode of operations:

- ISO 8613 sections 1 thru 7: Office Document Architecture and Interchange Format.
- ISO 8824 : Specification of Abstract Syntax Notation One (ASN.1).
- ISO 8325 : Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1).

1.2 *Introduction*

1.2.1 *Document architecture concept*

1.2.1.1 For the purpose of this recommendation, a document is defined as an amount of text that is interchanged between systems.

1.2.1.2 A document can be interchanged for two major purposes:

- It may be interchanged as an original in a final form allowing for printing, displaying and storing by the recipient,
- It may be interchanged in a revisable form allowing for processing by the recipient.

Processing includes editing, reformatting, filing and other manipulations.

1.2.1.3 Text is information for human comprehension that can be presented in a two-dimensional form, e.g. printed on paper or displayed on a screen.

1.2.1.4 Text consists of graphic elements such as character box elements, geometric elements and photographic elements, which constitute the content of a document.

1.2.1.5 The contents of a document can be separated into various portions in order to:

- delimit presentation objects such as pages,
- delimit logical objects such as paragraphs,
- use different types of coding,
- allow processing after communication.

The description of these portions of text and their relationship constitute the document architecture.

The architecture supports incorporation of a set of sub-architectures within the content, which are in accordance with other Recommendations (e.g. T.6).

1.2.1.6 The document architecture recognizes two structures:

- the layout structure,
- the logical structure.

The layout structure relates the content portions to layout objects for their positioning and rendition on the presentation media.

The logical structure relates the content portions to logical text objects serving specific purposes: sections, headings, paragraphs, footnotes, and figures.

The architecture that is particular to a given document is called specific document architecture.

1.2.1.7 A document contains a document profile as a set of attributes at document level. The document profile contains information for handling the document as a whole, and may repeat information in the document content.

1.2.2 *Coding schemes available*

- 1.2.2.1 Recommendation T.6 defines facsimile coding techniques to be used for photographic coded text.

The use of the end of facsimile block (EOFB) control function defined in Recommendation T.6 is mandatory at the end of each text unit of T.6 encoded data.

Note - Documents structured in accordance with this recommendation do not contain tiles encoded using the optional uncompressed mode of Recommendation T.6.

Note - The bit ordering of groups of 8 bits of T.6 encoded data into octets is as follows: The first bit of a group is placed into the LSB (bit 1), and the trailing bit of a group is placed into the MSB (bit 8). This is in accordance with T.70 section 5.5.1.2.

- 1.2.2.2 ISO 8613/7 defines facsimile encoding techniques for bitmap coded text.

- 1.2.2.3 The use of other coding techniques is for further study.

1.2.3 *Pel transmission density for photographic text*

- 1.2.3.1 Systems must provide the capability to interchange photographic coded text using a pel transmission density of 200 pels per 25.4 mm in both the horizontal and vertical directions.

- 1.2.3.2 Optional pel transmission densities may be negotiated (see Table 11/TTG/87-42).

1.2.4 *Orientation of tiled facsimile pages*

The intended viewing orientation of tiled facsimile pages may be either vertical or horizontal.

1.2.5 *Definitions*

- 1.2.5.1 Terms and their definitions are listed in Annex A.

- 1.2.5.2 Some of the terms used in this recommendation have been defined in ways that may differ from the meanings of similar terms in other standards.

1.3 *General characteristics of the interchange format*

1.3.1 *General*

- 1.3.1.1 Systems supporting the tiled facsimile mode of document interchange shall provide a minimum set of facilities specified herein.

1.3.1.2 These systems may provide other facilities in addition to the minimum set defined here. These facilities are negotiated separately from the set of facilities defined herein.

1.3.2 *Minimum set of facilities required for systems supporting tiled facsimile*

The minimum set of facilities required for systems supporting the tiled facsimile mode of document interchange are:

- 1.3.2.1 the ability to create and interchange documents in the tiled facsimile raster graphic content interchange format (see § 5.3);
- 1.3.2.2 the ability to position and dimension layout objects using a standardized coordinate system (see § 2.4.1);
- 1.3.2.3 the ability to designate the type of coding used to represent text contained in a block;
- 1.3.2.4 the ability to handle the maximum interchanged image area and to provide, at least, the image area for assured reproduction which are defined for North American paper sizes A thru K and ISO paper sizes A4 thru A0 (see § 2.4.4.2);
- 1.3.2.5 the ability to interchange documents, composed of:
 - a) one page containing only tiles encoded by the facsimile coding scheme defined in Recommendation T.6;
 - b) one page containing only tiles encoded by the bitmap encoding scheme defined in ISO 8613/7;
 - c) one page containing any combination of tiles encoded by the facsimile coding scheme defined in Recommendation T.6, or the bitmap encoding scheme defined in ISO 8613/7;
- 1.3.2.6 the ability to process up to 980 interchanged tiles for presentation as a single page, without using negotiation;
- 1.3.2.7 the ability to interchange images with a pel transmission density of 200 pels per 25.4 mm.;
- 1.3.2.8 the ability to interchange images with a fixed tile size of 512 pels x 512 pels.

1.3.3 *Negotiable facilities for tiled facsimile document interchange*

The procedure used for the negotiation of facilities is outside the scope of this standard.

Note - Negotiation is inherently more difficult for those cases where encoding is performed prior to the interchange of information between systems.

One or more additional facilities listed in this section may be provided by a system supporting tiled facsimile document interchange:

1.3.3.1 the ability to provide image areas for other paper sizes (for further study).

1.3.3.2 the ability to interchange images with other pel transmission densities;

1.3.3.3 the ability to interchange images with other tile sizes (for further study).

2 *Functions for the structuring and interchange of text*

2.1 *Document profile*

The document profile is a set of attributes at the highest level in the document structure. It provides supplementary information to facilitate handling the document as a whole.

2.2 *Specific layout structure*

The specific layout structure is a tree structure with a number of hierarchical levels.

The nodes of the tree are specific layout objects. The branches of the tree represent the division of specific layout objects into subordinate specific layout objects.

At the highest level of the tree is the specific document. At successively lower levels of the hierarchy are specific page(s) and specific block(s).

In the context of this recommendation, a page is defined as a rectangular area that corresponds to the interchanged image area. The page is the reference area used for positioning and imaging the content of the document. The size of the interchanged image area may be smaller than, equal to, or greater than the size of the corresponding physical page.

All layout objects subordinate to a page are positioned, directly or indirectly, relative to the page and are dimensioned such that they do not extend beyond the page.

A block is a rectangular area with its sides parallel to the sides of the page. It is a basic container for a portion of the document content.

All blocks are positioned relative to the next higher level of the layout hierarchy.

A tile is a block in a page in which all blocks have the same dimensions, and no part of any block may overlap any other block. They are positioned in a fixed grid, determined by partitioning the page into tile-sized areas. All tiles have the same pel path and line progression.

2.3 *Document interchange*

2.3.1 *General*

The interchange representation of a document in image form consists of a sequence of protocol elements, each representing the document profile, a layout object or a content portion. Three types of such elements are defined: a document profile descriptor, which represents the document profile; layout descriptors, which represent layout objects; and text units, which represent content portions. The order in which these descriptors and text units appear in the sequence of protocol elements is specified for each application in the corresponding application rules (see § 5).

2.3.2 *Document profile descriptor*

The document profile descriptor is a data element of the document interchange protocol, which consists of a sequence of subordinate data elements and elementary data items.

It consists of a sequence of data elements which represents the presentation capabilities a system must provide to be able to handle the document.

The elementary data items used to represent the document profile descriptor are data types such as numeric strings, octet strings, character strings, and bit strings.

2.3.3 *Layout descriptor*

A layout descriptor is an element of the document interchange protocol that represents a specific layout object and its attributes. Each layout object is represented by one layout descriptor.

A layout descriptor is a data element consisting of a sequence of subordinate data elements and elementary data items, each representing one attribute of the layout object. Each subordinate data element again consists of a sequence of subordinate data element and/or elementary data items. The elementary data items have basic data types such as numbers, character strings and bit strings.

2.3.4 *Text units*

A text unit is an element of the document interchange protocol that represents a portion of document content and the associated attributes.

A text unit is a data element that consists of two main parts:

- a) a sequence of data elements representing the attributes of the portion of document content.
- b) a sequence of one or more data elements representing the portion of document content itself.

The data elements used to represent the content have basic data types such as numeric strings, octet strings, character strings, and bit strings.

2.4 *Positioning of layout objects*

2.4.1 *Page coordinate system*

The position of all layout objects subordinate to pages are specified directly or indirectly by means of an orthogonal page coordinate system. The origin of this co-ordinate system is at the top left corner of the page. The horizontal axis (x axis) corresponds to the top edge and the vertical axis (y axis) corresponds to the left edge of the page. Horizontal positions are measured positively from the vertical axis to the right and vertical positions are measured positively from the horizontal axis downwards. All dimensions and positions within a basic layout object are specified as integral multiples of Basic Measurement Units (BMUs). All relative directions are expressed as counter clockwise angles of rotation with respect to some specified reference direction.

2.4.2 *Positioning of layout objects on a page*

The reference point for positioning is the top left corner of each layout object. A layout object at any level of the hierarchy is positioned relative to the reference point of the layout object at the next higher level and is contained entirely within the area of that layout object. Thus, the layout objects immediately below the level of page are positioned in absolute page coordinates, while all objects subordinate to that level use relative positioning.

2.4.3 *Positioning of text within a tile*

In positioning text within a tile, the area of the tile is treated as a sub-page that is independent of adjoining areas. The text image is not permitted to extend beyond the area of the tile.

2.4.3.1 *Pel path, line progression, and initial point*

Pel path is the direction of progression of successive pels along a line and is expressed as a direction relative to the horizontal axis of the page coordinate system. It may take one of four possible values: 0, 90, 180, and 270 degrees.

Line progression is the direction of progression of successive lines and is expressed as a direction relative to the pel path. It may take one of two possible values: 90 or 270 degrees.

The initial point is the point relative to which all imaged pels are positioned within the basic layout object. The first pel on the first line of the pel array is positioned at the initial point. Subsequent lines are positioned such that the first pel on each line falls in the direction of the line progression. Table 1/TTG/87-42 specifies the position of the initial point for various combinations of pel path and line progression, and Figure 2/TTG/87-42 provides two examples.

Table 1/TTG/87-42 Position of the Initial Point			
Pel Path	Line Progression	Horizontal Coordinate	Vertical Coordinate
0	270	0	0
0	90	0	BDV
270	270	BDH	0
270	90	0	0
180	270	BDH	BDV
180	90	BDH	0
90	270	0	BDV
90	90	BDH	BDV

where BDV = Vertical dimension of the block
BDH = Horizontal dimension of the block

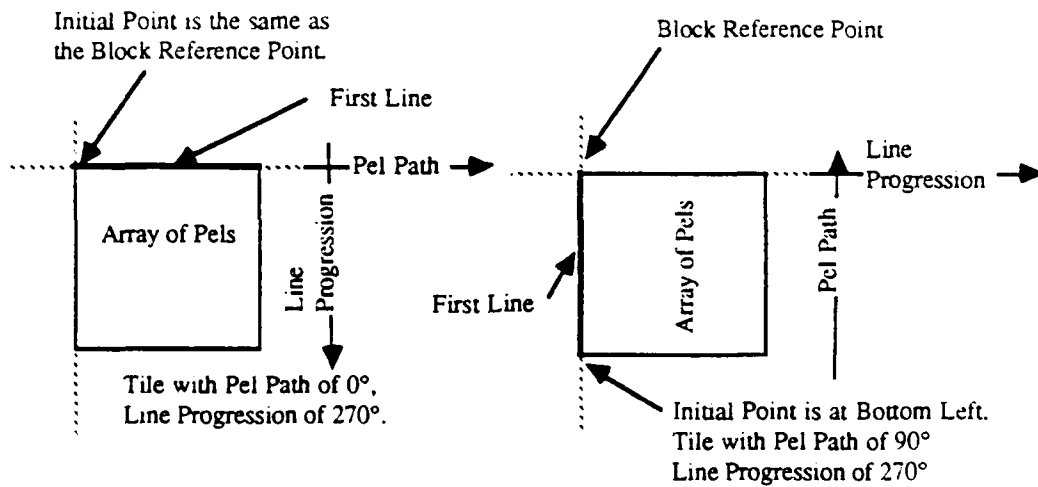


Figure 2/TTG/87-42
Position of pels in the block

2.4.4 *Dimensions for text presentation*

2.4.4.1 *Basic measurement unit (BMU)*

The size of the basic measurement unit (BMU) is 1/1200 x 25.4 mm.

2.4.4.2 *Paper sizes*

Different physical paper sizes can be used for presentation of facsimile information. Such paper sizes are ISO A4 (210 x 297 mm), North American A paper size (215.9 x 279.4 mm), other ISO A-series paper sizes up to A0, and other North American series paper sizes up to K size. The standard nominal paper sizes are listed in Table 2/TTG/87-42.

The use of other nominal page sizes (e.g. corresponding to other physical paper sizes) must be negotiated.

Table 2/TTG/87-42
Parameters for facsimile images for various page sizes

Paper Size	Dimensions (W x L)			Area of Assured Reproduction		
	BMU	mm	inches	BMU	mm	inches
A4	9920 x 14030	210 x 297		9240 x 13200	195.6 x 279.4	
A3	14030 x 19840	297 x 420		13340 x 19000	282.6 x 402.4	
A2	19840 x 28060	420 x 594		19160 x 27220	405.6 x 576.4	
A1	28060 x 39680	594 x 841		27380 x 38840	579.5 822.3	
A0	39680 x 56120	841 x 1189		39000 x 55280	825.5 x 1170.3	
A	10200 x 13200		8.5 x 11	9240 x 12400		7.70 x 10.33
North American Legal	10200 x 16800		8.5 x 14	9240 x 16000		7.70 x 13.33
B	13200 x 20400		11 x 17	12520 x 19560		10.43 x 16.30
C	20400 x 26400		17 x 22	19720 x 25560		16.43 x 21.30
D	26400 x 40800		22 x 34	25720 x 39960		21.43 x 33.30
E	40800 x 52800		34 x 44	40120 x 51960		33.43 x 43.30
F	33600 x 48000		28 40	32760 x 47160		27.3 x 39.3
G	13200 x 108000		11 90	12520 x 107160		10.43 x 89.3
H	33600 x 171600		28 143	32760 x 170760		27.3 x 142.3
J	40800 x 211200		34 176	39960 x 210360		33.3 x 175.3
K	48000 x 171600		40 143	47160 x 170760		39.3 x 142.3

2.4.4.3 Interchanged image area

In the context of this recommendation, a page is a rectangular area that corresponds to the interchanged image area. The page is a layout object that is used as the reference area for positioning and imaging the text information content. The page is intended to be positioned and imaged on a unit of the presentation surface. The ideal size of the presentation surface is a rectangular area called the nominal page. Thus the page is positioned on a single nominal page. The dimensions of the nominal page are determined by the attribute "medium type".

2.4.4.4 Image area for assured reproduction

When the interchanged image area uses the maximum page sizes specified in § 2.4.4.2, the possibility of edge losses must be considered when the text information is to be printed on paper. These edge losses may be caused, for example, by tolerances on the physical paper size, and by equipment tolerances.

Examples of image areas for assured reproduction are illustrated in Figure 3/TTG/87-42 and Figure 4/TTG/87-42, showing the maximum edge losses on each paper edge. The indicated edge losses are based on the idealized or nominal paper sizes as defined in § 2.4.4.2. See Table 2/TTG/87-42 for information on other nominal page sizes.

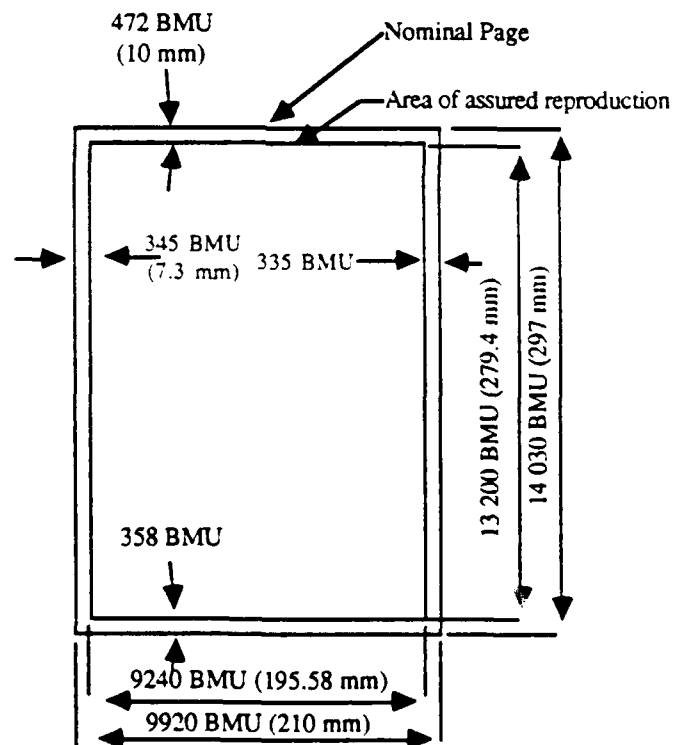


Figure 3/TTG/87-42
Nominal page area and area
of assured reproduction of the ISO A4 page size

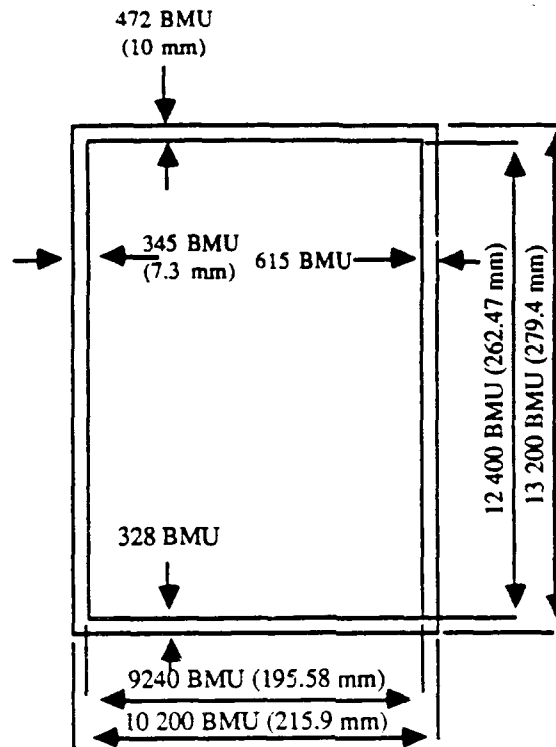


Figure 4/TTG/87-42
Maximum interchanged image area and area of assured reproduction of the North American letter page size

2.4.4.5 Positioning of the page relative to the nominal page

The reference point for the positioning of a page is the top left corner of the page. The position of the page reference point relative to the top left corner of the nominal page can be specified by the attribute "page position". In the absence of a specified page position, the following rules apply:

- a) When the dimensions of the page are equal to the dimensions of the nominal page, then the intended position of the page has been fully specified. That is, the reference point of the page is coincident with the top left corner of the nominal page and the edges of the page are coincident with the edges of the nominal page.
- b) When the dimensions of the page are greater than the dimensions of the nominal page, then the nominal page will be centered on the page.
- c) When the dimensions of the page are not equal to the dimensions of the nominal page, the recipient should center the page so as to minimize the possibility of loss of information.
- d) When the dimensions of the interchanged image are less than or equal to the dimensions of the assured reproduction area, the reference point of the page is coincident with the top left corner of the assured reproduction area.

2.5 Attributes

2.5.1 *Attribute classification*

Attributes are parameters of the specific layout objects and of the content portions that specify characteristics of and relationships between the objects and the content portions.

Categories of attributes of layout objects are:

- positioning attributes, specifying the dimensions and positions of the objects;
- presentation attributes, specifying how the content of the objects is to be imaged, e.g. resolution;
- user readable comments;
- pragma attributes, providing possibly redundant information which is provided by the originator and may either be used by the recipient to improve efficiency or may be safely ignored by the recipient.

Attributes of content portions are the type of coding and coding attributes.

The attributes of specific layout objects are defined in § 2.5.3 for positioning attributes, and in § 2.5.4 for presentation attributes. The attributes of content portions are defined in § 2.5.5.

2.5.2 *Default attributes*

Certain attributes, for example the positioning and presentation attributes of layout objects, can be specified either explicitly for the object to which they apply directly, or at higher levels of the hierarchy. In the latter case, the attributes are interpreted as default values for the lower levels. They can be overridden by specific attributes at the lower levels.

For example, it is possible to specify the default page size at document level, or the default resolution for photographic blocks at page level.

In addition, standard default values (to be used when no particular values are specified) are defined in this recommendation.

To determine the attributes of a specific layout object, the priority order is:

- 1) attributes specified explicitly in the specific layout object concerned;
- 2) the default attributes specified in the specific object at the next higher level, unless the specific object concerned is the document itself. These attributes may, in turn, be "inherited" from still higher levels in the specific structure;
- 3) the default values defined in this recommendation.

2.5.3 *Attributes of layout objects*

The attributes applicable to specific layout objects are defined in this section. Some attributes apply only to certain types of objects. Where this is the case, it is mentioned in the definition. Otherwise, the attribute applies to all object types.

2.5.3.1 *Object type*

This attribute specifies whether the object concerned is a document, a page, or a block.

This attribute is represented as a data element in the descriptor to which it applies.

2.5.3.2 *Object identifier*

This attribute identifies an object description uniquely within the context of the document.

An object identifier consists of a sequence of numbers. Each number in the sequence corresponds to a hierarchical level of the specific layout structure and identifies one particular object description at that level.

The numbers in this sequence start with the number corresponding to the document layout root object description. This is followed by each of the numbers corresponding to the object descriptions or the path through the hierarchical structure from the document layout root to the object description.

The first number in the sequence indicates that the identifier pertains to a layout object description. The value assigned to this first number is "1". An object identifier consisting of just this first number identifies the object description of the document layout root.

The actual value of each subsequent number is not significant; however the sequence of numbers allocated to each object description shall be chosen so that each object description can be uniquely distinguished from all other object descriptions in the document.

The object identifier is represented by a string of decimal-coded numerals with a "space" character as a separator between each pair of successive numerals.

2.5.3.3 *User-readable comments*

This attribute contains a character sequence that is to be interpreted as comments relevant to that object or to the associated content portions. This character sequence is not part of the body of the document.

This attribute is represented as a data element in the descriptor of the object to which it applies. The contents of this data element consists of a sequence of characters coded according to ANS X3.4.

2.5.3.4 *Default value lists*

This attribute specifies a set of attribute value lists that are applicable as defaults to subordinate objects of the designated object types. Use of such a list is not permitted if it is specified as being applicable to the same or a higher level than the object in which it appears.

Each such list is represented as a data element in the contents of a descriptor. Each such list must identify the object type to which it is to be applied: i.e., either page or block. The contents of each such list consists of a set of attribute data elements that specify the default values to be applied.

A default value list may specify the following attributes:

- all positioning attributes,
- all presentation attributes,
- pragma attributes.

2.5.3.5 *Position*

This attribute applies only to blocks. It consists of a pair of coordinates that specify the position of the block relative to the containing page.

This attribute is represented as a data element that is applicable to the level of block. The content of this data element consists of two integers that specify the *X* and *Y* coordinate distances in basic measurement units (BMUs), from the positioning reference point of the containing page to the reference point of the block to which this attribute applies.

2.5.3.6 *Dimensions*

This attribute applies only to pages and blocks. It consists of a pair of dimensions that specify the size of the object.

This attribute is represented as a data element that is applicable to the level of page, or block. The content of this data element consists of two integers that specify the *X* and *Y* dimensions of the object in basic measurement units (BMUs).

2.5.3.7 *Pragma attributes*

Three attributes are provided at the page level which may optionally be used to enhance performance:

a) *Layout covers entire page*

This attribute indicates that a block exists for every tile location.

b) *Tile index*

This attribute supports the accessing of selected tiles in a large image.

c) *Interchange - Ordered*

This attribute indicates that interchange ordering of the blocks is the same as the ordering of the index information.

2.5.4 *Presentation attributes*

This is a category of attributes of objects at the lowest level of the layout hierarchy.

2.5.4.1 *Content type*

This attribute identifies the type(s) of graphic elements forming the content of the object. In doing so, it also designates the use of the set of presentation attributes that may be applied to that content type.

In the initial applications defined in this recommendation, the value of the content type attribute indicates the use of photographic elements. Other content types are for further study.

2.5.4.2 *Attributes of photographic element content type*

2.5.4.2.1 *Image orientation attributes*

a) *Pel path*

This attribute specifies the direction of progression of successive photographic elements along a line.

b) *Line progression*

This attribute specifies the direction of progression of successive lines, relative to the direction of the graphic element (pel) path, as defined in § 2.4.3.1.

2.5.4.2.2 *Image resolution attributes*

a) *Pel transmission density*

This attribute specifies the number of photographic elements (pels) per unit of length in both the vertical and horizontal directions.

In the absence of negotiation, this attribute is represented as a data element that designates a value of 200 pels per 25.4 mm.

2.5.5 *Attributes of content portions*

2.5.5.1 *Type of coding*

This attribute specifies the coding used to represent the content. In the initial applications of this recommendation, photographic elements are coded either according to CCITT Recommendation T.6 or according to the bitmap coding specified in ISO 8613/7. Other types of coding are for further study.

This attribute is represented as a data element in a text unit.

2.5.5.2 *Coding attributes*

These attributes are associated with the type of coding of the content portion. They provide additional parametric information used in encoding/decoding the content portion.

These attributes are represented as data elements in a text unit.

In the case of photographic elements, the following attributes are defined:

a) *Number of pels per line*

This is specified as an integer data element.

The value is fixed at 512.

3 *Functions for the interchange of text in processable form*

For further study.

4 *Specifications of protocol elements*

This section provides the detailed specifications of the protocol elements introduced elsewhere in this recommendation.

The protocol elements are represented using the presentation transfer syntax of Abstract Syntax Notation 1 (ASN.1) as defined in ISO 8824 and 8825, and are specified in this section using the corresponding notation. An overview of ISO 8824 and 8825 is contained in Annex B.

The formal definitions of all protocol elements and their subordinate data elements are presented in Figures 5 to 11/TTG/87-42. The following restrictions apply to these definitions and their use in the document interchange protocol:

- Whenever a "named number list" is specified in connection with the data type "integer", all values not explicitly listed are reserved for future assignment.
- All application-wide and context-specific data element identifiers that are not assigned in Figures 5/TTG/87-42 to 11/TTG/87-42 are reserved for future assignment.
- Length fields longer than five octets shall not be used in applications of this recommendation. A length field of five octets allows for the presentation of a length of up to 4,294,967,295.

4.1 *Protocol elements*

The following protocol elements are used in this recommendation: document profile descriptor, layout descriptor and text unit. These are formally defined in Figure 5/TTG/87-42.

4.2 *Document profile descriptor*

The document profile descriptor represents the document profile and includes a data element for presentation capabilities. The resulting specification is contained in Figure 6/TTG/87-42.

4.3 *Document characteristics descriptor*

The document characteristics descriptor is used in the negotiation and invocation of the presentation capabilities (Figure 7/TTG/87-42). It includes the basic terminal characteristics, the interchange format and the non-basic capabilities. The latter contain sequences of non-basic attribute values used in the document.

4.4 *Layout descriptor*

As specified in Figure 8/TTG/87-42, a layout descriptor is a sequence of two components. The first of these indicates which type of layout object the descriptor represents, namely document, page or block. The second component contains any other attributes of the object. It takes the form of a set drawn from the following: dimensions, presentation attributes, default value lists for subordinate objects, user-readable comments and pragmas.

The following data types are introduced to facilitate the definitions of the positioning, dimensioning and presentation attributes:

Measure — used to represent coordinate distances or dimensions, in units of BMU, that may be specified as either fixed or variable.

Measure pair — used to represent pairs of coordinate distances or dimensions, in units of BMU.

4.4.1 *Default value list*

One set of layout attributes is defined for each object type, namely page and block. This choice is specified in Figure 9/TTG/87-42. Common definitions are provided for attributes that apply to more than one object type. Three attributes or groups of attributes are specified: position, dimensions and presentation attributes.

4.4.2 *Presentation attributes*

As specified in Figure 10/TTG/87-42, the presentation attributes include content type and photographic attributes.

The following data types are introduced to facilitate the definitions of the photographic attributes:

One of four angles — used to represent angles of 0, 90, 180, or 270 degrees.

One of two angles — used to represent angles of 90 or 270 degrees.

4.5 *Text units*

As specified in Figure 11/TTG/87-42, a text unit is a sequence of two components. The second component is the text information itself, whereas the first one includes any attributes of the content portion that the text unit represents. Content portion attributes are: type of coding, coding attributes and alternative graphic representation.

Interchange data unit	::= CHOICE {
documentProfile	[0] IMPLICIT DocumentProfileDescriptor,
layoutObject	[2] IMPLICIT LayoutObjectDescriptor,
contentPortion	[3] IMPLICIT TextUnit}

Figure 5/TTG/87-42
Formal definition of protocol element

DocumentProfileDescriptor	::= SET {
documentCharacteristics	[2] IMPLICIT documentCharacteristics OPTIONAL}

Figure 6/TTG/87-42
Formal definition of document profile descriptor

```

document Characteristics ::= SET (
  application Profile      CHOICE {
    documentArchitectureLevel [0] IMPLICIT INTEGER {tiled-facsimile(4)( Note 1)},
    contentArchitectures     [1] IMPLICIT INTEGER {fda-1(1)} OPTIONAL,
    TRF-0                   [5] IMPLICIT SET{
    interchangeFormatLevel   [10] IMPLICIT OCTET STRING},
    NonBasicStructuralCharacteristics [6] IMPLICIT INTEGER {if-a1(0)} OPTIONAL,
    [3] IMPLICIT SET{
    numberOfObjectsPerPage   [0] IMPLICIT INTEGER OPTIONAL} OPTIONAL)

```

Figure 7/TTG/87-42

Formal definition of document characteristics

Note 1: This value was temporarily assigned by TTG until an official assignment has been made.

```

LayoutDescriptor ::= SEQUENCE (
  layoutObjectType      LayoutObjectType,
  layoutDescriptorBody LayoutDescriptorBody OPTIONAL)

LayoutObjectType ::= INTEGER (documentLayoutRoot (0), page (2), block (4))

LayoutDescriptorBody ::= SET (
  referencesToSubordinateObjects CHOICE {
    [0] IMPLICIT SEQUENCE OF NumericString,
    [1] IMPLICIT SEQUENCE OF NumericString} OPTIONAL,
    position [3] IMPLICIT MeasurePair OPTIONAL,
    dimensions [4] IMPLICIT MeasurePair OPTIONAL,
    presentationAttributes [6] IMPLICIT PresentationAttributes OPTIONAL,
    defaultValueLists [7] IMPLICIT SEQUENCE OF DefaultValueList OPTIONAL,
    userReadableComments [8] IMPLICIT CommentString OPTIONAL,
    pagePosition [15] IMPLICIT MeasurePair OPTIONAL,
    mediumType [16] IMPLICIT MediumType OPTIONAL,
    tileIndex [24] IMPLICIT IndexOfTiles OPTIONAL ( Note 2),
    layoutCoversEntirePage [25] IMPLICIT BOOLEAN OPTIONAL ( Note 2),
    interchangeOrdered [26] IMPLICIT BOOLEAN OPTIONAL ( Note 2)

MeasurePair ::= SEQUENCE (Measure, Measure)

Measure ::= CHOICE (
  fixedMeasure [0] IMPLICIT INTEGER)

CommentString ::= X3.4 String
— same character set as PrintableString
— plus carriage return and line feed

MediumType ::= SEQUENCE(
  nominalPageSize MeasurePair OPTIONAL)

IndexOfTiles ::= SEQUENCE (
  pe1DirectionTileCount INTEGER,
  lineDirectionTileCount INTEGER,
  encodings INTEGER {allT6(0), allBitmap(1), mixed(2)} OPTIONAL ( Note 2),
  textUnitAddressList SEQUENCE OF FixedLengthInteger (see Note 1 ),
  textUnitContentAddressList SEQUENCE OF FixedLengthInteger (see Note 1 ) OPTIONAL)

```

Figure 8/TTG/87-42
Formal definition of layout descriptor

Note 1: FixedLengthInteger is a private data type encoded identically to an INTEGER, with the restriction that all FixedLengthInteger objects have 32 bits of data, and allowing 9 or more leading bits of all 0's or 1's.

Note 2: These value was temporarily assigned by TTG until an official assignment has been made.

DefaultValueList	::= CHOICE {
pageAttributes	[2] IMPLICIT PageAttributes,
blockAttributes	[4] IMPLICIT BlockAttributes}
PageAttributes	::= SET {
dimensions	<Attribute OPTIONAL,
presentationAttributes	<Attribute OPTIONAL}
BlockAttributes	::= SET {
position	<Attribute OPTIONAL,
dimensions	<Attribute OPTIONAL,
presentationAttributes	<Attribute OPTIONAL}
Attribute	::= CHOICE {
position	[0] IMPLICIT MeasurePair,
dimensions	[1] IMPLICIT MeasurePair,
presentationAttributes	[3] IMPLICIT PresentationAttributes}

Figure 9/TTG/87-42
Formal definition of default value list

PresentationAttributes	::= SET {
rasterGraphicsAttributes	[1] IMPLICIT RasterGraphicsAttributes OPTIONAL}
RasterGraphicsAttributes	::= SET {
pelPath	[0] IMPLICIT OneOfFourAngles OPTIONAL,
lineProgression	[1] IMPLICIT OneOfTwoAngles OPTIONAL,
pelTransmissionDensity	[2] IMPLICIT PelTransmissionDensity OPTIONAL}
OneOfFourAngles	::= INTEGER {d0 (0), d90 (1), d180 (2), d270 (3)}
OneOfTwoAngles	::= INTEGER {d90 (1), d270 (3)}
PelTransmissionDensity	::= INTEGER {p180 (0), p200 (1), p240 (2), p300 (3), p400 (4), p600 (5), p1200 (6)}

Figure 10/TTG/87-42
Formal definition of presentation attributes

TextUnit	::= SEQUENCE {
contentPortionAttributes	ContentPortionAttributes OPTIONAL,
contentInformation	OCTET STRING}
ContentPortionAttributes	::= SET {
contentIdentifierLayout	ContentPortionIdentifier OPTIONAL,
typeOfCoding	[0] IMPLICIT TypeOfCoding OPTIONAL,
codingAttributes	CHOICE {
rasterGraphicsAttributes	[2] IMPLICIT RasterGraphicsAttributes} OPTIONAL}
TypeOfCoding	::= INTEGER { t6 (1), bitmap(2)}
RasterGraphicsAttributes	::= SET {
numberOfPelsPerLine	[0] IMPLICIT INTEGER OPTIONAL}

Figure 11/TTG/87-42
Formal definition of text unit

5 *Application rules*

5.1 *Interchange formats*

The interchange format to be used for documents consisting of one page of tiled photographic elements is Tiled Raster Format 0 (TRF.0).

The specification of other interchange formats is for further study.

5.2 *Order of transmission*

The transmission of the description of a layout structure follows the natural order, i.e. the description of the tree structure (other forms of description are for further study).

The text units follow the description of the layout structure.

The order of transmission of the descriptors and text units is illustrated in Figure 12/TTG/87-42.

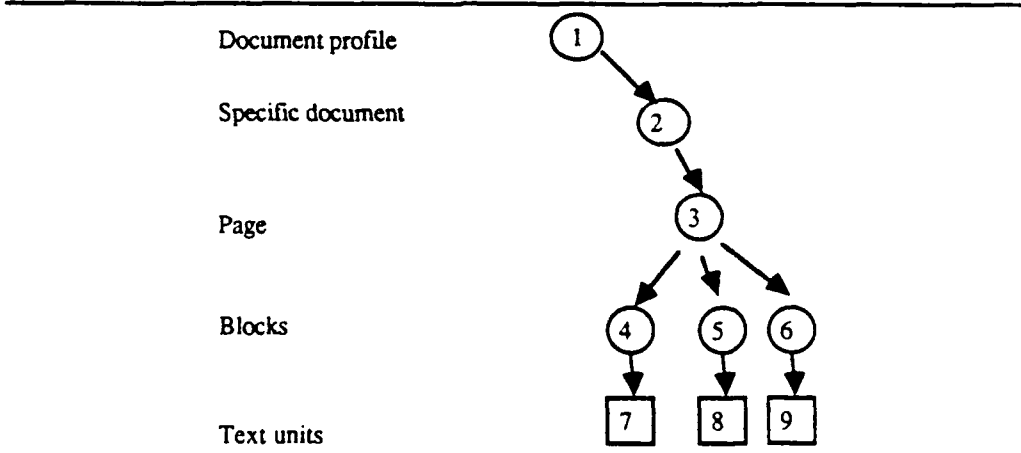


Figure 12/TTG/87-42
Transmission sequence of protocol elements

5.3 *Interchange format TRF.0*

This format is defined with a specific layout structure consisting of:

- i) document descriptor,
- ii) page descriptor,
- iii) block descriptor(s),
- iv) text unit(s).

There is only one page per document. Generics are not allowed. All blocks are tiles. The page is an integral number of tiles in each dimension. Tiles are allowed to be absent. If a tile is absent, the result is as if the background was cleared prior to the imaging of the document.

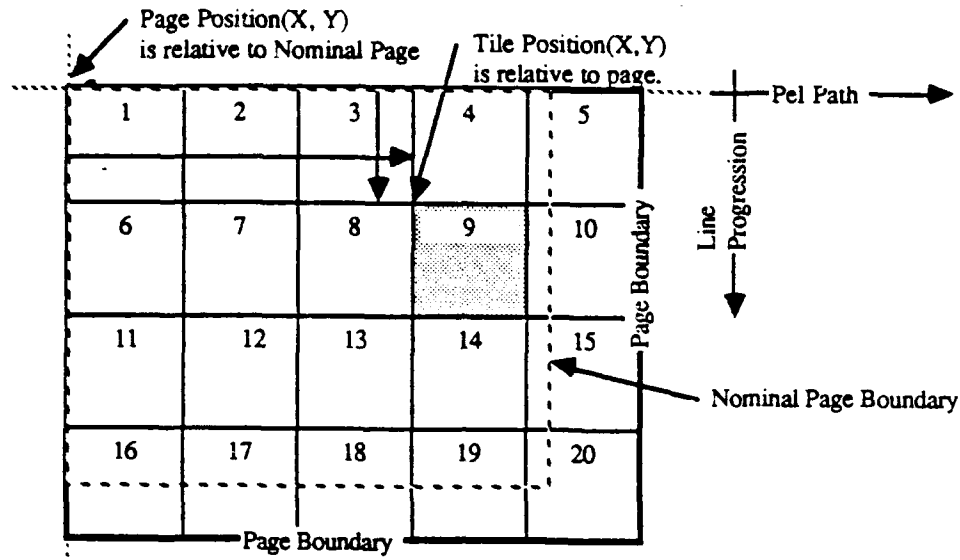
Some pragmas are defined as follows:

The first pragma, "Layout covers entire page", is a Boolean which indicates whether or not all tiles are present. This allows the recipient to avoid having to clear the image prior to imaging a document.

The second pragma, "Tile Index", is a data structure which contains text unit addresses. It may also optionally contain text unit content addresses. This allows the recipient to achieve more rapid random access.

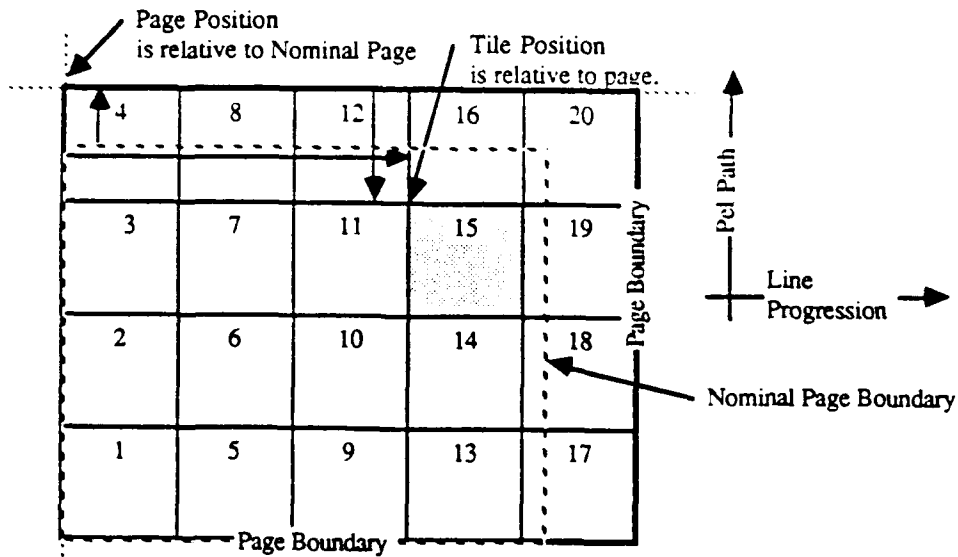
The text unit address list and the text unit content address list each contain an entry for every possible tile in the pel path order of the containing page. If the tile is not present, the corresponding entry is a zero address; otherwise, the entry contains the appropriate address. If any entry is zero, the "Layout covers entire page" pragma is required to be false if present. Page-relative addresses in the text unit content address list refer to the primitive octet string of the content portion of the text unit.

"Interchange-Ordered" is an optional Boolean defined for the tile index. If true, it indicates that the block interchange order matches the index order. If false, no relationship between the orders is indicated (see Figures 13-TTG/87-42 and 14-TTG/87-42).



Note: Numbers indicate the implicit Tile order based on Pel Path and Line Progression of un-rotated tiles.

Figure 13-TTG/87-42
Page and Tile Positions for an Unrotated Image.



Note: Numbers indicate the implicit tile order based on Pel Path and Line Progression of unrotated tiles.

Figure 14-TTG/87-42
Page and Tile Positions For an Image With a
Viewing Orientation at 90° from the Scanning Orientation

- 5.3.1 Every descriptor or text unit comprises a number of attributes, listed in Tables 3/TTG/87-42 to 10/TTG/87-42, that are all required.

In these tables, descriptors and their attributes as well as presentation attributes are listed. The attributes are diversely qualified:

- m: Mandatory, indicates the attribute must be explicitly stated at each occurrence;
 d: Defaultable, indicates the attribute is always necessary, but may be described elsewhere according to the default mechanisms, as defined in § 2.5.2;
 nm: Non-mandatory, indicates the attribute is not always used by the sender, depending upon the specific needs.

Table 3/TTG/87-42

TRF.0 — Attributes of the document characteristics descriptor

List of attributes	Category	Defined values	Standard default values
Document characteristics			
Application profile	m	Tiled Facsimile	
Content architecture	m	TRF.0	
Non-basic structural characteristics	nm		

Table 4/TTG/87-42
TRF.0 — Attributes of layout descriptors

List of attributes	Category	Defined values	Standard default values
Document descriptor			
Object type	m	Document	
Object identifier	nm		
Default value lists	nm	Page, block	
User readable comment	nm	X 3.4 coded string	
Page descriptor			
Object type	m	Page	
Object identifier	nm		
Reference to subordinate objects	nm	Block Identifier	
Medium typenm	See Table 5/TTG/87-42		
Presentation attributes	d	See Table 7/TTG/87-42	
Layout covers entire page	nm	Boolean	False
Interchange-Ordered	nm	Boolean	False
Page size	m	Width, Height in BMU (Note 2)	
Page position		nmX, Y BMU (Note 4)	
Default value lists	nm	Block	
Tile index	nm	See Table 6/TTG/87-42	
Block descriptor	(Note 1)		
Object type	m	Block	
Object identifier	nm		
Position	m	X, Y BMU (Note 3)	
Dimensions	d	Width = W BMU (Note 5) Height = H BMU (Note 5)	
Reference to content portions	nm	Content portion identifier	

Note 1 — The text unit for a block is required to be a primitive octet string.

Note 2 — Page size must be an integral multiple of block size on each axis.

Note 3— The position of the block must correspond to a valid tile location. The tiles are laid out in a fixed grid covering the page.

Note 4— If the dimensions of the page are equal to or greater than the dimensions of the nominal page, the default is to center the image on the nominal page. If the dimensions of the page are less than those of the nominal page, the default is to image the document with the reference point of the page placed coincident with the top, left corner of the assured reproduction area.

Note 5 — Width = W BMU and Height = H BMU, where W and H are equal to :
 $512 * (1200 + (\text{pel transmission density in pels per } 25.4 \text{ mm}))$.

Table 5/TTG/87-42

TRF.0 — Attributes of Medium Type (per ISO 8613/2 § 14.2.1.4)

List of attributes	Category	Defined values	Standard default values
Nominal page size	nm	Width, Height BMU	Width = 9920 BMU Height = 14 030 BMU (ISO A4)

Table 6/TTG/87-42

TRF.0 — Attributes of index of tile locations

List of attributes	Category	Defined values	Standard default values
Pel direction tile count	m	Integer > 0 (Note 1)	
Line direction tile count	m	Integer > 0 (Note 1)	
Interchange ordered	nm	Boolean	
Text Unit Address List	m	Sequence of fixedLengthInteger	
Text Unit Content Address List	nm	Sequence of fixedLengthInteger	

Note 1 — Each tile is a block with the dimensions specified in Table 4/TTG/87-42. The tiles are laid out in a fixed grid which covers the page.

Table 7/TTG/87-42

TRF.0 — Attributes of the text unit

List of attributes	Category	Defined values	Standard default values
Content portion identifier	nm		
Type of coding	d	Photographic: T.6 (Note 1) Bitmap: ISO 8613/7	Photographic: T.6
Coding attributes for photographic information			
Number of pels per line	d	512	
Pel array order	d	Up, Down (Note 2)	Down

Note 1 — TRF.0 text units with T.6 photographic encoding do not include data encoded per the uncompressed mode of T.6.

Note 2 -- This attribute is only applicable if the value of the attribute "Type of Coding" is 'bitmap encoding'. 'Down' specifies the allocation of the first pel to the most significant bit, and 'up' specifies the reverse order.

Table 8/TTG/87-22

TRF.0 — Presentation attributes for photographic elements

List of attributes	Category	Defined values	Standard default values
Content type	m	Photographic	
Pel path	d	0°, 90°, 180°, 270°	0°
Line progression	d	270°, 90°	270°
Pel transmission density	d	200 pels per 25.4 mm	200 pels per 25.4 mm

Table 9/TTG/87-42

TRF.0 — Attributes of document profile descriptor

List of attributes	Category	Defined values
Non-basic structural characteristics	nm	

Table 10/TTG/87-42

TRF.0 — Attributes of Non-basic structural characteristics

List of attributes	Category	Defined values
Maximum number of objects per page	nm (Note 1)	≤980

Note 1 -- The number of objects per page is the maximum number of objects per page in the interchanged data.

5.4 Values for negotiable facilities for TRF.0

This section refers to options that are not necessarily implemented and therefore shall be the subject of a negotiation prior to document interchange.

Table 11/TTG/87-42

TRF.0 — Presentation attributes for photographic elements

List of attributes	Category	Values for negotiable facilities
Pel transmission density	nm	240, 300, 400, 600, 1200 pels per 25.4 mm

ANNEX A

(to Recommendation TTG/87-42)

Terms and definitions

Note — Some of the terms used in this recommendation have been defined in ways that may differ from the meanings of similar terms in other recommendations.

<i>Term</i>	<i>Paragraph</i>	<i>Term</i>	<i>Paragraph</i>
Area of assured reproduction	A.1	Interchange format	A.37
Attribute	A.1a	Layout directive	A.38
Basic object	A.2	Layout object	A.39
Basic measurement unit (BMU)	A.3	Layout structure	A.40
Block (text block)	A.4	Length, length indicator	A.41
Character	A.5	Length field	A.42
Character base line	A.6	Logical object	A.43
Character box	A.7	Logical structure	A.44
Character box element	A.8	Mixed mode	A.45
Character path	A.9	Nominal page	A.45a
Composite object	A.10	Office document architecture (ODA)	A.46
Constituent	A.11	Overlay	A.47
Constructor	A.12	Overscan	A.47a
Content	A.13, A.15	Page	A.48
Content portion	A.14	Page-relative address	A.49
Control function	A.16	Parameter	A.50
Data element	A.17	Pel path	A.51
Data structure	A.18	Photographic coded text	A.52
Descriptor	A.19	Photographic element	A.53
Document	A.20	Physical page	A.53a
Document: body	A.21	Pictorial character	A.53b
Document class	A.22	Portion of text	A.54
Document descriptor	A.23	Pragma	A.55
Document profile	A.24	Presentation	A.56
Document structure	A.25	Presentation medium	A.57
Editing	A.26	Processing	A.58
Enveloping data	A.27	Rendition	A.59
Frame	A.28	Retrieval	A.60
Generic	A.29	Specific	A.61
Geometric element	A.30	Symbol	A.62
Graphic element	A.31	Text	A.63
Graphic element (of text)	A.32	Text image format (TIF)	A.64
Image	A.33	Text processable format (TPF)	A.65
Image area	A.34	Text unit	A.66
Information field	A.35	Tile	A.67
Interchange	A.36		

A.1 area of assured reproduction

The area of an image guaranteed to be reproduced when edge losses due to equipment or paper tolerances are considered.

A.1a attribute

A property of a document or a constituent of a document expressing a characteristic of the document or constituent concerned, or a relationship with one or more other documents or constituents.

Note — In recommendations T.61 and T.62, related properties and characteristics of devices as well as presentation attributes are called parameters.

A.2 basic object

An object that is not subdivided.

Note — A basic object may be structured internally according to a presentation architecture.

A.3 basic measurement unit (BMU)

A unit of measurement used for positioning and dimensioning of layout objects. The size of the basic measurement unit is 1/1200 x 25.4 mm.

A.4 block (text block)

A basic layout object corresponding to a rectangular area within a page or within a frame with its sides parallel to the sides of the enclosing page or frame, in which only one category of graphic element is to be imaged.

A.5 character

A member of a set of elements (upon which agreement has been reached and that is) used for the organization, control or representation of information (data).

A.6 character base line

A positioning reference for the placement of symbols within a character box.

A.7 character box

1) A rectangular area on the presentation medium that can be used for the rendition of one graphic character.

2) A rectangular area within which a graphic character is contained. The nominal spacing between symbols, if any, is included within the character box.

A.8 character box element

Either language characters or pictorial characters that are presented within character boxes.

Note — This term is also as an alternative term for graphic character.

A.9 character path

The direction of progression of successive character boxes along a line.

A.10 composite object

An object that is subdivided into other composite objects and/or basic objects.

Note — Composite layout objects are termed document or page.

A.11 constituent

Tiled Raster Interchange Format / TRIF 1.0

TTG/87-42

A layout or logical object below the level of document or a content portion.

A.12 constructor

A data element whose content is itself a data element, or a series of data elements.

A.13 content

The actual information conveyed by the document, independent of layout structure and logical structure.

A.14 content portion

A part of the content of the document associated with at most one basic layout object and one basic logical object.

Note — Other terms in use are "portion of text" and "portion of document content".

A.15 contents

Substance of a data element containing the primary information the data element is intended to convey.

Note — Sometimes the contents is termed "value".

A.16 control function

An action that affects the recording, processing, transmission, or interpretation of data and that has a coded representation consisting of one or more octets of bits.

A.17 data element

A sequence of data elements serves for the coded representation of a document or its constituents. A data element consists of three components that always appear in the following order: identifier, length, indicator, contents.

Note — See also *data structure*.

A.18 data structure

A set of data items representing an object, a content portion, a part of an object or content portion, or the document description. The data items constituting a data structure represent attributes of the constituents or the document description concerned.

Note — The term *data structure* might be replaced by a *data element* or *constructor* (element).

A.19 descriptor

A data element representing a layout object or a logical object.

A.20 document

An amount of text that can be interchanged as a unit defined by the originator between applications.

A.21 document body

The content of a document and the attributes of the layout and logical objects, excluding the document profile.

A.22 **document class**

A category of document defined by a set of common properties, e.g. letter, memorandum, report invoice.

A.23 **document descriptor**

A set of attributes describing the layout or logical structure of a document.

A.24 **document profile**

A set of attributes associated with a document, for the purpose of handling the document as a whole.

A.25 **document structure**

The result of dividing and subdividing the content of a document into increasingly smaller parts, the parts being called layout objects, logical objects, and text units.

A.26 **editing**

The carrying out of operations associated with altering the content of a document, e.g. replace, insert, delete.

A.27 **enveloping data**

Information added to a document to ensure its interchange.

A.28 **frame**

A composite layout object within a page or within another frame with its sides parallel to the sides of the enclosing page or frame, intermediate at one or more levels between the page and the block in a layout structure.

A.29 **generic**

Term qualifying a layout or logical structure, a layout or logical object, or an attribute pertaining to a document class.

A.30 **geometric element**

A collection of drawing primitives including points, arcs, lines, rectangles, and polygons used to construct drawing in a prescribed area.

A.31 **graphic element**

A character, other than a control function, that has a visual representation normally handwritten, printed, or displayed. Graphic characters include simple alphanumeric characters, composite characters (e.g. accented letters) and pictorial characters (e.g. mosaics).

A.32 **graphic element (of text)**

The smallest individually specified element used to construct an image. There are three categories of graphic elements of text, namely character box elements, geometric elements and photographic elements.

- A.33 **image**
Visual presentation of a text.
- A.34 **image area**
That part of a page that is available for assured reproduction of text.
- A.35 **information field**
Part of a text unit that contains the content portions (i.e. the textual information).
- A.36 **interchange**
The process of providing a duplicate of a document to a receiving person or device.
- A.37 **interchange format**
A representation of a document by a collection of data elements for the purpose of interchange.
- A.38 **layout directive**
An attribute of a logical object that specifies the manner of presentation (e.g. rendition, positioning), optionally in relation to a layout object.
- A.39 **layout object**
One of the parts pertaining to the layout structure, e.g. page, block.
- A.40 **layout structure**
The result of dividing and subdividing the content of a document into increasingly smaller parts, on the basis of the presentation, e.g. into pages and blocks.
- A.41 **length, length indicator**
Component of a data element specifying the length of the contents in octets.
- A.42 **length field**
The field in a data element that contains the length indicator.
- A.43 **logical object**
One of the parts pertaining to the logical structure, e.g., chapter, section, paragraph.
- A.44 **logical structure**
The result of dividing and subdividing the content of a document into increasingly smaller parts, on the basis of the meaning of the content, e.g. into chapters, sections, paragraphs.
- A.45 **mixed mode**

A mixed mode capability provides the means of transferring the information content of a document between sender and recipient, where the information content has been encoded using different techniques (e.g. in all forms of facsimile or character coding) and the document structure fully identified enabling the recipient to apply sophisticated editing methods.

A.45a nominal page

The area of an image corresponding to paper sizes ISO A4 through A0 and North American paper sizes A through K (see table 2/TTG/87-42)

A.46 office document architecture (ODA)

Rules for applying structure to office documents.

A.47 overlay

Positioning of layout objects in such a manner that they overlap each other partially or fully on the presentation medium.

A.47a overscan

Photographic elements outside the nominal page area. Overscan is used to allow for the repositioning of text such that it falls within the nominal page area..

A.48 page

A layout object that is a rectangular area with dimensions equal to the associated interchanged image area.

A.49 page-relative address

The page-relative address of an object is the difference, expressed as the number of octets, between the position of the first octet of the object and the position of the first octet of the layout descriptor for the containing page.

A.50 parameter

A term sometimes used instead of attribute.

A.51 pel path

The direction of progression of successive photographic elements along a line.

A.52 photographic coded text

Text represented using photographic elements.

A.53 photographic element

An individual picture element (pel, pixel) used in arrays to construct images. Each pel has a specific shape, size, colour, intensity and position.

A.53a physical page

A piece of paper or an electronic display

A.53b pictorial character

Predetermined pattern which is intended to be presented in adjacent character boxes to construct rulings, boxes, figures, logos, diagrams, or other pictures occupying multiple character boxes.

A.54 **portion of text**

See content portion.

A.55 **pragma**

Auxiliary information which may either be used as guidance in order to improve efficiency or may be ignored.

A.56 **presentation**

The printing or display of stored graphic element to allow for human comprehension of the stored information.

A.57 **presentation medium**

The carrier of information in a form perceptible to a human, e.g. a sheet of paper or a display screen.

A.58 **processing**

The carrying out of operations on a document. This includes editing, formatting, rendition, filing, retrieval.

A.59 **rendition**

The operation consisting of presenting the document content on a presentation medium.

A.60 **retrieval**

The recovery of previously filed information.

A.61 **specific**

Term qualifying a layout or logical structure, a layout or logical object, or an attribute, pertaining to a particular document.

A.62 **symbol**

See graphic element.

A.63 **text**

Text is information for human comprehension that is intended for presentation in two-dimensional form, e.g. printed on paper or displayed on a screen. Text consists of symbols, phrases, or sentences in natural or artificial languages, pictures, diagrams and tables.

A.64 **text image format (TIF)**

An interchange format that provides for the representation of the document profile, layout objects and content portions of a document.

A.65 **text processable format (TPF)**

An interchange format that provides for the representation of the document profile, logical objects, content portions and, optionally, layout objects of a document.

A.66 **text unit**

A data structure representing a content portion.

A.67 **tile**

A tile is a block in a page in which:

- all blocks have the same dimensions
- all blocks have the same pel path and line progression
- no portion of any block overlaps any other block
- the X and Y coordinates of each block position are integral multiples of the respective block dimensions.

ANNEX B

(to recommendation TTG/87-42)

Summary of presentation transfer syntax

The protocol specified by this recommendation is based upon the transfer syntax defined in ISO 8824 and 8825. This annex briefly describes that syntax and some of the associated concepts, and illustrates its use by examples.

B.1 *Data types, data values and standard notation*

ISO 8824 and 8825 define a transfer syntax for various kinds of information. Each piece of information is considered to have a type as well as a value. A *data type* is a class of information (for example, numeric or textual). A *data value* is an instance of such a class (for example, a particular number or a fragment of text). ISO 8824 and 8825 define a number of generally useful data types from which application-specific data types are constructed in this recommendation and in others that make use of the ISO 8824 and 8825 transfer syntax. Among the generally useful data types defined by ISO 8824 and 8825 are Integer, Octet, String, Sequence and Set.

The *standard notation* defined in ISO 8824 and 8825 is a formal description method that allows data types relevant for an application to be specified in terms of other data types, including the generally useful data types of ISO 8824 and 8825. This notation is used in § 5 of the present recommendation, where the protocol element data types are specified in terms of Sets and Sequences of more elementary data types which in turn are specified in terms of others, and finally in terms of basic data types such as Integer and Octet String.

B.2 *Standard representation*

The *standard representation* for a data type is the set of rules for encoding values of that type for transmission as a sequence of octets. The representation of a value also encodes its type and length, and is completely implied by the standard notation of the data type.

The standard representation of a data value is a *data element* having three components, which always appear in the following order. The *Identifier* designates the data type and governs the interpretation of the Contents. The *Length* specifies the length of the Contents. The *Contents* is the substance of the object, containing the primary information the object is intended to convey. The Identifier and the Length each consist of one or more octets; the Contents consists of zero or more octets.

B.2.1 *Identifier*

Four *classes* of data types are distinguished by means of the Identifier: universal, application-wide, context-specific and private-use. *Universal* types are generally useful, application-independent types; they are defined in ISO 8824 and 8825. *Application-wide* types are more specialized, being peculiar to a particular application; they are defined in this recommendation, and in others using ISO 8824 and 8825, by means of the standard notation. *Context-specific* types, like application-wide types, are peculiar to an application and defined using the standard notation. However, they are used only within an even more limited context — for example, that of a Set — and their identifiers are assigned so as to be distinct only within that limited context. *Private-use* types are reserved for private use; the assignment of specific private-use Identifiers can be accomplished by means of the standard notation but is outside the scope of ISO 8824 and 8825 and this recommendation.

Two forms of data elements are distinguished by means of the Identifier: primitive and constructor. A *primitive* element is one whose Contents is atomic. A *constructor* element is one whose Contents is itself a data element, or a series of data elements. Constructor elements are thus recursively defined.

B.2.2 Length

The Length specifies the length in octets L of the Contents and is itself variable in length. It may take any of three forms: short, long and indefinite.

The *short* form may (but need not) be used when L is less than 128.

The *long* form must not be longer than five octets as specified in TTG/87-42 § 4.

The *indefinite* form may (but need not) be used when the element is a constructor. When this form is employed, a special *end-of-contents* (EOC) element terminates the Contents.

Note — All constructor data elements, whether or not of indefinite length, are ultimately composed of primitive data elements (perhaps with several intervening "levels" of constructor data elements). Primitive data elements *always* have a definite length.

B.2.3 Contents

The Contents are variable in length and are interpreted in a type-dependent way. If the data element is a constructor, the Contents themselves comprise zero or more elements; data elements are thus recursively defined.

B.3 Built-in types and defined types

The generally useful data types defined by ISO 8824 and 8825 consist of built-in types and defined types.

Built-in types are used to construct all other data types. They include Integer, Octet String, Sequence, Set and Tagged. Integer is a primitive data type. Octet String can be either primitive or constructor. Sequence and Set are constructor data types. Identifiers for these data types are of the universal class and are specified in ISO 8824 and 8825. A Tagged data type is a data type for which the Identifier can be specified using the standard notation, as is done in § 5 of this recommendation.

Defined types are specified in ISO 8824 and 8825 using the standard notation. They include Numeric String, Printable String, and Recommendation T.61 String, all of which are defined in terms of the built-in type Octet String. They can be either primitive or constructor; the identifiers are of the universal class and are specified in ISO 8824 and 8825.

ANNEX C

(to Recommendation TTG/87-42)

Summary of data element identifier assignment

This annex contains a collection of tables that summarizes the assignment of the data element identifiers specified in § 5.

Each table corresponds to a SEQUENCE or SET defined in § 5. The table specifies the identifiers of all data elements that may occur within the SEQUENCE or SET concerned. Each row of the table specifies:

- 1) the value of the identifier in hexadecimal notation (including the bits representing the class and the form of the identifier);
- 2) the implied built-in data type (only SEQUENCE, SET, INTEGER and OCTET STRING are implied by the definitions in § 5);
- 3) the name of the data element (with proper word separation and capitalization).

Context: TTG/87-42 (the protocol)

Identifier	Implied data type	Data element name
A0	SET	Document profile descriptor
A2	SEQUENCE	Layout descriptor
A3	SEQUENCE	Content portion

Context: Document profile descriptor (SET)

Identifier	Implied data type	Data element name
81	OCTET STRING	Reference to specific layout structure
A2	SET	Document Characteristics

Context: Document characteristics (SET)

Identifier	Implied data type	Data element name
80	INTEGER	Application profile
A3	SET	Non-basic structural characteristics
86	INTEGER	Interchange format level (if-a1 (0))
81	INTEGER	Document architecture level (fda-1 (1))
A5	SET	Content Architectures

Context: Non-basic structural capabilities (SET)

Identifier	Implied data type	Data element name
80	INTEGER	Number of objects per page

Context: Content architecture (SET)

Identifier	Implied data type	Data element name
81	CHOICE	TRF-0 (value = 10) (Note1)

Note 1: These values were temporarily assigned by TTG until an official assignment has been made.

Context: Presentation attributes (SEQUENCE) in presentation capabilities

Identifier	Implied data type	Data element name
89	INTEGER	Pel path
8A	INTEGER	Photographic line progression
8B	INTEGER	Pel transmission density

 Context: Layout descriptor (SEQUENCE)

Identifier	Implied data type	Data element name
02	INTEGER	Layout object type
31	SET	Layout descriptor body

 Context: Layout descriptor body (SET)

Identifier	Implied data type	Data element name
41	OCTET STRING	Object identifier
A3	SEQUENCE	Position (Measure pair)
A4	SEQUENCE	Dimensions (Measure pair)
A6	SET	Presentation attributes
A7	SEQUENCE	Default value lists
88	OCTET STRING	User-readable comments
AF	SEQUENCE	Page position (Measure pair)
B0	SET	Medium type
B8	SEQUENCE	Tile index (<i>Note 1</i>)
99	BOOLEAN	Layout covers entire page (<i>Note 1</i>)

Note 1: These values were temporarily assigned by TTG until an official assignment has been made.

 Context: Measure pair (SEQUENCE)

Identifier	Implied data type	Data element name
80	SEQUENCE	Fixed measure (Horiz), Fixed measure (Vert)

 Context: Fixed measure

Identifier	Implied data type	Data element name
80	INTEGER	Measurement

Context: Medium type (SEQUENCE)

Identifier	Implied data type	Data element name
A0	SEQUENCE	Nominal page size (Measure pair)

Context: Default value lists (SEQUENCE)

Identifier	Implied data type	Data element name
A2	SET	Page attributes
A4	SET	Block attributes

Context: Tile index (SEQUENCE) (Note 1)

Identifier	Implied data type	Data element name
80	INTEGER	PeI direction tile count
81	INTEGER	Line direction tile count
82	BOOLEAN	Interchange-Ordered
83	INTEGER	Encodings
A2	SEQUENCE	Reference to block descriptor
A3	SEQUENCE	Reference to content of text unit

Note 1: These values were temporarily assigned by TTG until an official assignment has been made.

Context: Page attributes (SET)

Identifier	Implied data type	Data element name
A1	SEQUENCE	Dimensions
A3	SET	Presentation attributes

Context: Block attributes (SET)

Identifier	Implied data type	Data element name
41	OCTET STRING	Reference to content portions
A0	SEQUENCE	Position
A1	SEQUENCE	Dimensions
A3	SET	Presentation attributes

Context: Presentation attributes (SET) in layout descriptor

Identifier	Implied data type	Data element name
A1	SET	Photographic attributes

Context: Photographic attributes (SET)

Identifier	Implied data type	Data element name
80	INTEGER	Pel path

TTG/87-42

Tiled Raster Interchange Format / TRIF 1.0

81	INTEGER	Line progression
82	INTEGER	Pel transmission density

Context: Text unit (SEQUENCE)

Identifier	Implied data type	Data element name
31	SET	Content portion attributes
04	OCTET STRING (primitive)	Text information

Context: Content portion attributes (SET)

Identifier	Implied data type	Data element name
41	OCTET STRING	Object identifier
80	INTEGER	Type of coding (2 for bitmap, 1 for T 6 Compressed)
A2	SET	Coding attributes (Raster graphics)

Context: Coding attributes (SET)

Identifier	Implied data type	Data element name
80	INTEGER	Number of pels per line
81	INTEGER	Number of lines

ANNEX D

(to Recommendation TTG/87-42)

Coding examples

This annex provides coding examples using the Tiled Raster Interchange Format (TRIF).

Example 1.

Description and encoding example using TRIF for an entirely white A4 size document.

Note numbers in parentheses refer to notes below.

document-profile {	A0 07
document-characteristics {	A2 05
content-architectures {	A5 03
raster-graphics-content-archs trf-0 (Note D)}}	81 01 0A (Note 2)
layout-object {	A2 48
object-type page,	02 01 02
descriptor-body {	31 43
dimensions {	A4 08
horizontal fixed 12288,	80 02 30 00
vertical fixed 15360}}	80 02 3C 00
presentation-attributes {	A6 0B
photographic-attributes {	A1 09
pel-path d0,	80 01 00
line-progression d270,	81 01 03
pel-transmission-density p6}}	82 01 01
default-value-list-layout {	A7 0C
block-attributes	A4 0A
dimensions {	A1 08
horizontal fixed 3072,	80 02 0C 00
vertical fixed 3072}}}}	80 02 0C 00
user-readable-comments 'TTG Example 1',	88 0D 54 54
	47 20 45 78
	61 6D 70 64
	65 20 31
medium-type {	B0 0A
nominal-page-size { (Note 3)	A0 08
horizontal fixed 9920,	80 02 26 C0
vertical fixed 14030}}	80 02 36 CE
layout-covers-entire-page TRUE}} (Note 4)	99 01 FF

layout-object {	A2 0D
object-type block,	02 01 04
descriptor-body {	31 08
position {	A3 06
horizontal fixed 3,	80 01 03
vertical fixed 3}}	80 01 03
content-portion {	A3 51
content-portion-attributes {	31 4E
type-of-coding t.6,	80 01 01 (Note 6)
raster-graphics-content-attributes {	A2 04
number-of-pels-per-line 512}}	80 02 02 00
content-information 'FFFFFF ... FFFFFFF000880'H)	04 43 FF FF
	...[62xFF]
	00 08 80 (Note 7)
/	
layout-object {	A2 0E
object-type block,	02 01 04
descriptor-body {	31 09
position {	A3 07
horizontal fixed 3075,	80 02 0C 03
vertical fixed 3}}	80 01 03
content-portion {	A3 51
content-portion-attributes {	31 4E
type-of-coding t.6,	80 01 01
raster-graphics-content-attributes {	A2 04
number-of-pels-per-line 512}}	80 02 02 00
content-information 'FFFFFF ... FFFFFFF000880'H)	04 43 FF FF
	...[62xFF]
	00 08 80
\	

Repeat previous block layout-object and content-portion (marked with "|")s, for each block, changing the position value for each block layout-object.

Note 1. -- Raster-graphics-content-archs is discussed in ISO/DIS 8613-7 Annex C. TTG has invented the classification trf-0.

Note 2. -- These values were temporarily assigned by TTG until an official assignment has been made.

Note 3. -- Nominal page size is from Table 2/TTG/87-42.

Note 4. -- Layout-covers-entire-page is a layout descriptor invented by TTG. The tag of 99H has been temporarily assigned.

Note 5. -- The position of the block is the coordinates of the center of the pixel at the upper left corner of the tile.

Note 6. The value of 01H to encode the t.6 type-of-coding was temporarily assigned by TTG until a value is assigned in ISO 8613-7.

Note 7. The bit ordering in this example has put groups of 8 bits of T.6 encoded data into octets by putting the first bit of a group into the LSB (bit 1) and trailing bit of a group into the MSB (bit 8).

ANNEX E**(to Recommendation TTG/87-42)****Tiling Task Group Goals and Intent**

This annex briefly describes some of the reasoning that went into the development of the interchange format and clarifies a few concepts.

E.1 Purpose

The need for an interchange format using a tiling scheme arose from a number of considerations associated with handling large format engineering drawings in raster format. In developing the tiling format, issues were raised and evaluated concerning the usefulness and efficiency of tiling as a technique for handling large format raster images between and within a wide range of systems. It was generally agreed that a tiling scheme could be developed that would receive broad, industry-wide support.

Therefore, it was the intent of the Tiling Task Group (TTG) to recommend a well-defined scheme to foster industry adoption. The TTG felt that adoption would be facilitated because most parameters were fixed, thereby simplifying implementation. Fixed parameters also support interchange of raster images on physical media.

The tiling scheme developed provides a format that supports operation on a subset of an image without requiring other portions of the image to be accessed. For large format documents this provides a way to interchange images between systems of various capabilities.

System efficiencies were a primary consideration in development of the tiling scheme and interchange format. Concerns associated with processing, system cost, real-time access and archival storage were discussed and considered. A tile format was developed for interchange that could also reasonably be used for storage and retrieval without necessarily requiring translation.

E.2 Background

Interest in an industry-supported tiling scheme was first expressed at a meeting with DOD on April 15, 1987. As a result of that meeting, a task group composed of industry representatives including system integrators, peripheral manufacturers, and users, as well as government representatives, assembled in an open forum to exchange views on tiling. This ad hoc group (TTG) concluded that development of an interchange format using tiling was desirable. Subsequently, a number of meetings and reviews were held by the TTG in order to develop this standard.

E.3 Methodology

It was the intent of the TTG to develop a standard that used existing and emerging standards as a basis wherever possible. This strategy assured that efforts were within the mainstream of raster imaging standards and promoted interoperability with other raster graphics formats utilized in related office document architecture standards.

A number of potential capabilities were identified which could have an impact on the interchange format. These were all evaluated and considered. Where appropriate, some of these capabilities were incorporated, others were reserved for future consideration, and still others were intentionally excluded.

In an effort to keep the interchange as simple as possible, a number of parameters such as tile size were fixed. It was decided that the only capability to remain negotiable between interchanging parties was resolution. Media and database issues were considered to be outside the realm of this standard and were specifically excluded from consideration.

E.4 Excluded Features

This interchange format deals only with bi-tonal (black and white) data. Pixels are assumed to be square.

A tile is a block in a page in which all blocks have the same dimensions and no part of any block overlaps any other block. They are positioned in a fixed grid, determined by partitioning the page into tile-sized areas.

For the purposes of this standard, all tiles are blocks and all blocks are tiles. All tiles are square. Tiles are allowed to be absent.

A single tile size is desirable to limit the burden on implementors of the interchange standard. The tile size is specifically 512 by 512 pels. This size was chosen as a compromise between the line buffer memory requirements of larger tiles and the storage overhead burden of smaller tiles. While it was recognized that system implementations exist that use 256 pel or 1024 pel square tiles, all vendors present agreed that a migration to 512 pel square tiles was acceptable for interchange.

The standard was developed within the framework provided by the existing document architectures work, especially the ISO 8613 series of documents. Much of the work was to identify a sufficient subset of the existing architecture specification which would lend itself to a tiling scheme. Ease of implementation was a primary consideration.

This standard borrows freely from existing standards, with certain restrictions and logical extensions. With the exception of pragmas, the extensions were all within the range of negotiable items in the existing standards. The pragmas are optional extensions which contain possibly redundant information used for efficient implementation. The Abstract Syntax Notation is used throughout the standard.

A non-exhaustive list of restrictions follows: Only one page (one single raster image) is allowed per document. The page is an integral number of tiles in each dimension. Unused constructs include frames, generic objects, page set, logical structures and objects and styles.

Blocks are restricted to be tiles, as defined above. Transparency is not used. The encoding scheme is a subset of data stream A, as found in ISO 8613/5.

Parameters which are handled as negotiable facilities in facsimile cause problems for an interchange format which is primarily intended for use on physical media. The TTG handled this by reducing the number of negotiable options and by extending the concept of negotiation. To use negotiable options in a transfer of documents, organizations need to make arrangements by means outside the interchange standard. The only negotiable item is resolution.

Any given tile is only to be encoded as T.6 compressed data or as ISO 8613 bitmap data. Each T.6 compressed tile ends in a EOFB as specified in T.6. Where T.6 encoding is used, the uncompressed escape option defined there is not supported. This is done because it places an unreasonable burden on implementors, is not used by any known vendors, and is not supported by existing or planned VLSI. This escape option is rendered unnecessary by the ability to insert bitmap encoded data per ISO 8613 on a tile-by-tile basis.

There is a one-to-one correspondence between block descriptors and text units.

There may not be more than one tile imaged to a given tile location. Every tile that exists is imaged to a single, unique tile location.

The case where a tile location has no associated tile is allowed (so-called "null tiles"). Tiles are allowed to be missing. In the event a tile is missing, the resulting image is to be imaged as if that tile is all white.

An upper limit is set on the number of tiles allowed in an image. A limit of some sort allows vendors to put a bound on system memory requirements.

The page-level attribute of tile rotation permits a raster editor to rotate all individual tiles in place; that is, without changing the order of tiles within the file. All tiles in a page have the same orientation, however.

Pragma is a term introduced from software terminology to describe non-required information which can convey to the sophisticated implementor information which a less sophisticated implementor can safely ignore. An example is the flag which says that all tiles are arranged in a sequence parallel to the sequence of tile index entries. One can safely ignore this information and correctly image the file by using each tile's position field.

E.5 Alternatives

Issues considered and specifically excluded are variable tile sizes, rectangular (non-square) tiles, overlapping tiles, tiles of variable transparency, and multiple pages per document. These and other items could be addressed in a later, separate application profile, but there were strong feelings that they do not belong in a limited, minimalist interchange format.

This proposal defines raster file format only and not issues related to database management such as document information, aperture card Hollerith code, document and page relationships, sheets, revisions, and multiple aperture card frames. These issues are outside the scope of Tiled Raster Interchange Format and are being dealt with by other organizations.

ANNEX F

(to Recommendation TTG/87-42)

Application of the Tile Index

I. Aspects of Tile Indexing

1. Pel path order is the order in which the pixels and tiles were (or would have been) scanned.
2. Block transmission order is the order in which the blocks are transmitted or stored. See figure 12/TTG/87-42 in Section 5.2.
3. The page reference point is the top left corner when the page is viewed in natural viewing orientation.
4. Each block descriptor has the coordinates of the block with respect to the page.
5. The pixels within a block are imaged according to the pel path parameter. As a result, the block may undergo rotation when it is imaged.

II. Structure specified by TRIF

The tile index is an optional structure containing the follow elements:

1. the tile counts in the pel path direction and the line count direction;
2. an optional Boolean called "Interchange - Order" specifies whether or not the tile index order matches the block transmission order. Tile index order refers to the order of both the sequence of text unit addresses and to the order of the sequence of text unit content addresses. If the Boolean is false, no relation between the tile index order and the block transmission order may be construed;
3. an optional Boolean call "Encodings" specifies whether all tiles have the same type of coding. There are four possibilities: all T.6 encoded, all bitmap, mixed, and unspecified;
4. the sequence of fixed-length text unit addresses, one per possible tile. If a tile is absent, 0 is substituted for the address of text unit corresponding to the tile;
5. an optional parallel sequence of text unit content addresses, one per possible tile. If a tile is absent, 0 is substituted for the address of the primitive octet string of the content portion of the text unit for that tile.
6. An optional Boolean specifies whether every possible tile is present. (Layout covers entire page.)
7. The tile index order is always pel path order.
8. Pel paths of all tiles have the same orientation.

III. Lazy Rotation

Lazy rotation is a logical rotation, i.e. the bits are not physically rotated within the file. Rather, the blocks are "renumbered" but not actually rotated, except when imaged. Specifically,

1. The block position coordinates change.
2. The pel path maintains its relationship to the blocks, not to the stationary background.
3. The page reference point maintains its relationships to the new top left corner of the nominal page, not to the blocks.
4. When the blocks are imaged, they are rotated in place according to the pel path.

As a result,

1. If the index order was pel path order prior to the rotation, it correctly remains in pel path order after the rotation, without needing any shuffling.
2. The block transmission order does not change due to the rotation.

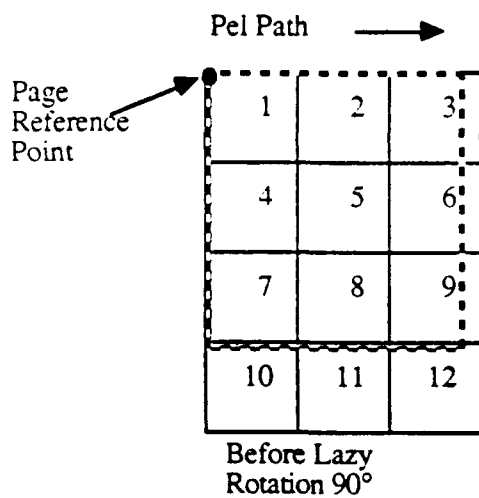
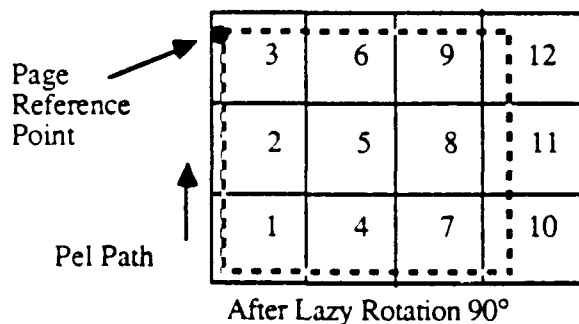


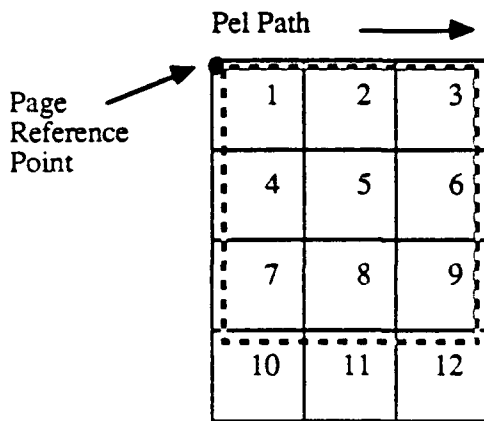
Figure shows conceptualization of disk image. Dotted line shows nominal page. Note that in all the figures below, numerals shown are intended to be actual bitmap images.
Index Order: 1, 2, 3, 4, 5, 6,
Block transmission Order: 1, 2, 3, 4, 5, 6, ...



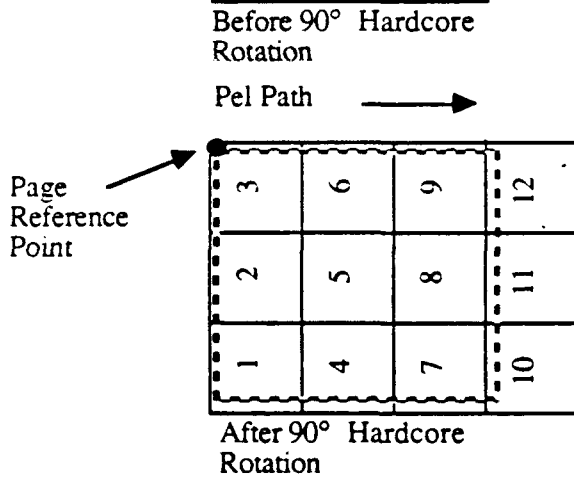
Only the block position coordinates have changed.
Index Order: 1, 2, 3, 4, 5, 6,
Block transmission Order: 1, 2, 3, 4, 5, 6, ...

IV. Rescanned Rotation

Rescanned rotation is a complete physical rotation. The end result looks exactly like the drawing was rotated and rescanned. The file index must be shuffled in order to maintain pel path ordering. The text units must also be shuffled if maintaining parallel ordering between the tile index and the block transmission order is desired.



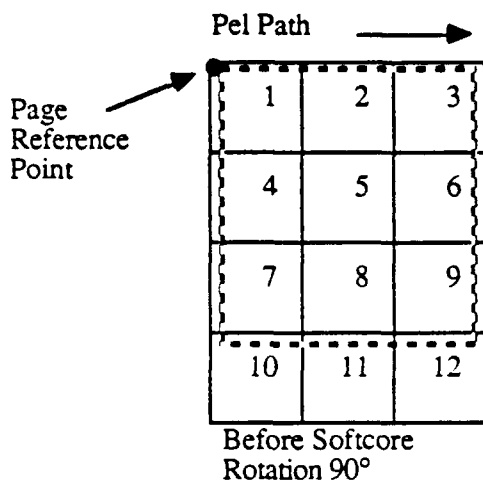
Index Order: 1, 2, 3, 4, 5, 6,
 Block transmission Order: 1, 2, 3, 4, 5, 6, ...



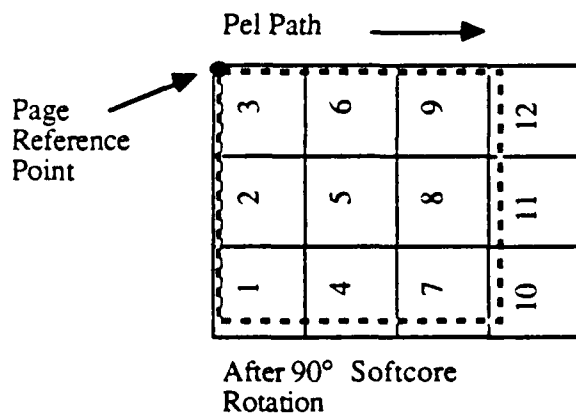
Relationship between stationary background and page reference point does not change. Block position coordinates changed. Blocks rotated physically in place.
 Index Order before shuffling: 1, 2, 3, ...
 Index Order after shuffling: 3, 6, 9, ...
 Block transmission order before shuffling: 1, 2, 3, 4, ...
 Block transmission Order after shuffling: 3, 6, 9, ..

V. Shuffled Rotation

Shuffled rotation is a physical rotation identical to hardcore rotation except that the blocks are not shuffled. The result of the rotation is that tile index is shuffled in order to maintain pel path order, but now the order of the index does not match the block transmission order since the blocks were not rotated.



Index Order: 1, 2, 3, 4, 5, 6,
 Block transmission order: 1, 2, 3, 4, 5, 6, ...



Relationship between stationary background and page reference point does not change. Block rotated physically in place.
 Index Order before shuffling: 1, 2, 3, ...
 Index Order after shuffling: 3, 6, 9, 12, ...
 Block transmission Order: 1, 2, 3, 4, 5, ...

VI. Why the "Interchange-order" bit is used.

Some implementors may not want to spend time processing headers looking for attribute information. They propose implementing imaging by using the "Layout covers entire page" pragma, the "Text Unit Address List", the "Text Unit Content Address List", the "Encodings" pragma, and the "Interchange-order" pragma.

There is a start-up cost, but once the two index-lists are in memory and if the two Boolean pragmas are true, and if the Encodings pragma indicates that all tiles have the same type of encoding, then imaging can be rapid. This may also allow the use of chained DMA devices rather than a general purpose processor in the imaging process. The locations and extents of the content portions are easily calculated, and no random access is necessary.

CONFORMANCE TESTING

NBS PLAN FOR VALIDATION (CONFORMANCE TESTING) OF COMPUTER
PRODUCTS IN SUPPORT OF THE CALS PROGRAM

NBS PLAN FOR VALIDATION (CONFORMANCE TESTING) OF COMPUTER
PRODUCTS IN SUPPORT OF THE CALS PROGRAM

of

June 2, 1987

Edited by
Sharon J. Kemmerer
NBS/ICST

TABLE OF CONTENTS

INTRODUCTION	1
PURPOSE	1
BACKGROUND	1
DISCUSSION	1
TERMS AND DEFINITIONS	2
ORGANIZATION	2
STRATEGY FOR DEVELOPING UNIVERSAL CONFORMANCE TESTING PROGRAMS	4
I. ACCREDITATION PROGRAM DEVELOPMENT	4
Criteria	4
Guide for Developing an Accrediting Program	4
Application for Accreditation	5
II. TESTING METHODS	5
Commonly accepted conformance test methods	5
Guidelines for development of test method approaches	5
General criteria for selecting appropriate test method	6
Test Suite Structure	8
Maintenance of Test Methods	9
Other Types of Standards Testing Methods	10
Interim procedures	10
III. CERTIFICATION AND REPORTING CONSIDERATIONS	10
Certification Body	10
NBS/ICST Role in Conformance Testing As Government Agent	12
Role of Standards Bodies (National and International)	13
Conformity Summary Report	13
Certificate of Conformity format, content and presentation	15
Applicability of Certificate of Validation	15
Types of Certification Systems	16
IV. TESTING ADMINISTRATION	16
General Guidelines for Test Method Administration	16
Criteria and procedures for Alternative Testing Methods	17
Associated costs	19
Resource and overhead requirements	19
Criteria and Procedures for Requesting Validation Services	19
V. TESTING LABORATORIES	22
National Laboratory Accreditation Programs	22
Conditions for Accreditation	23

Criteria for Accrediting Testing Laboratories . . .	24
Testing Laboratory Assessment	30
Responsibilities of Testing Laboratories	34
VI. MUTUAL RECOGNITION OF VALIDATION/CERTIFICATION . . .	35
Considerations and issues to be resolved	35
VII. CONFORMANCE TESTING REQUIREMENTS FOR PROGRAM APPLICATIONS	35
Establish procedures for testing specific applications	35
VIII. LEGAL ISSUES	36
Legal and copyright issues	36
IX. REFERENCES	36
APPENDICES:	
A. NBS PUBLICATIONS ON SOFTWARE STANDARDS	39
FEDERAL INFORMATION PROCESSING STANDARDS . . .	39
SPECIAL PUBLICATIONS AND OTHER REPORTS	40
B. CONFORMANCE TESTING STRATEGY FOR STANDARDS OF INTEREST TO CALS	43
INTRODUCTION	44
CONFORMANCE TESTING STRATEGY	45
STANDARDS HAVING CALS-SPECIFIC TESTING REQUIREMENTS	57
TESTING SITE CANDIDATES	60
INDUSTRY-FUNDED COOPERATIVE ARRANGEMENT	60
C. STANDARDS ORGANIZATIONS	61
D. TERMS AND DEFINITIONS	63
E. BOILERPLATE CERTIFICATE OF ACCREDITATION	69

"NBS PLAN FOR VALIDATION (CONFORMANCE TESTING) OF COMPUTER PRODUCTS IN SUPPORT OF THE CALS PROGRAM"

INTRODUCTION

I. PURPOSE.

To define the National Bureau of Standards (NBS) Program Plan for product conformance testing in support of computer standards required for CALS. Verification and acceptance testing will not be discussed under the body of this document.

II. BACKGROUND.

The Institute for Computer Sciences and Technology (ICST) at NBS is responsible for developing U.S. Government-wide standards for computer software, hardware, data management and networks. ICST works through voluntary industry standards organizations to develop standards that will meet the needs of Government users and be implemented in off-the-shelf commercial products. Standards that promise sizable benefits to the Government are issued as Federal Information Processing Standards (FIPS). Appendix A: "NBS Publications on Software Standards," provides more information on a subset of FIPS Publications and their availability.

Technically sound national and international standards are needed to preserve open competition in international markets and to support increased productivity and delivery of services at reduced cost. Standards provide users with off-the-shelf, compatible software products. Without standards, users cannot easily link products of different manufacturers together into systems and networks and fully utilize the training and skills of their staff. By promoting common understanding and expectations for products, standards also help to reduce risks and uncertainties in the marketplace.

Validation of computer products claiming conformance with standards further reduces risks and uncertainties to vendors and users; therefore, uniform testing procedures should be employed to perform this validation of conformance. NBS, through past experience in research and testing, sees a need for expansion of the efforts in structuring conformance testing. NBS can draw from expertise developed in such cooperative efforts as research associate programs and industry liaison, as well as promote in the national and international standards bodies, the ideas of a conformance testing strategy plan.

III. DISCUSSION.

The goal of this Strategic Plan is to provide an initial structure and approach for uniformity in conformance testing programs Government-wide. With Government-wide implementation, DoD and CALS communities will be able to benefit from the

information gathered and the "lessons learned" throughout the Government, and, at the same time, stay in synchrony with Government and industry-practiced testing procedures for commercial products.

This Plan does not currently hold all the answers. The Plan and individual conformance testing programs will be continually developed and enhanced beyond the completion of this CALS requirement, as various testing procedures are prototyped, and test cases are gathered, studied, and evaluated. For additional information specific to the CALS program, refer to Appendix B.

In the national and international standards arena, numerous discussions are taking place and work is underway in conformance test development for selected standards (Appendix C). This Plan describes what is currently happening and identifies some of the issues and timeframes of this development. The work is progressing too slowly for some standards pertinent to CALS, since no interested party or organization has taken the lead in development; therefore, CALS requirements will not necessarily be met automatically either in functionality or within the desired timeframes. OSD may see a need to invest resources and/or funding in support of the initial "general" conformance testing program development for high priority standards, as this Plan is intended to help NBS build interest in OSD, DoD, and the standards community.

Since NBS already has an internationally recognized program for accrediting testing laboratories (National Voluntary Laboratory Accreditation Program - NVLAP), NBS/ICST wants to stay aligned with the work that's already been done. For the benefit of the reader, an "*" precedes requirements already in use by NVLAP.

IV. TERMS AND DEFINITIONS. In order to ensure clarity when discussing conformance testing, terms and definitions used during discussion are provided in Appendix D.

V. ORGANIZATION.

The remainder of this Strategic Plan is the Strategy, and is organized in the following manner:

- I. ACCREDITATION PROGRAM DEVELOPMENT
- II. TESTING METHODS
- III. CERTIFICATION AND REPORTING CONSIDERATIONS
- IV. TESTING ADMINISTRATION
- V. TESTING LABORATORIES
- VI. MUTUAL RECOGNITION OF VALIDATION/CERTIFICATION
- VII. CONFORMANCE TESTING REQUIREMENTS FOR PROGRAM APPLICATIONS
- VIII. LEGAL ISSUES
- IX. REFERENCES

Appendices

- A. NBS Publications on Software Standards
- B. Conformance Testing Strategy for Standards of Interest to CALS
- C. Standards Organizations
- D. Terms and Definitions
- E. Boilerplate Certificate of Accreditation

STRATEGY FOR DEVELOPING UNIVERSAL CONFORMANCE TESTING PROGRAMS

I. ACCREDITATION PROGRAM DEVELOPMENT

A. Standards Identification. Criteria for identifying FIPS standards requiring conformance testing.

1. cost benefit
2. needs and scope of the user population
3. nature and content of other relevant conformance testing programs
4. importance of the FIPS to commerce, consumer well-being
5. the economic and technical feasibility of accrediting testing laboratories for the test methods, types of test methods, products, or services requested.
6. whether an administrative testing body is available

B. *Guide for Developing an Accrediting Program. The following issues need to be addressed in developing a new accreditation program:

1. Decide what test methods should be offered for accreditation.
2. Decide what the units of accreditation should be, i.e., should the test methods be offered individually or should they be logically grouped; should all test methods of a standard specification be offered as a group.)
3. Identify the critical elements (environmental/sample conditioning, test equipment and apparatus, procedures), for each test method, e.g., quality assurance checks.
4. Establish assessment techniques and priorities based on the complexity and importance of each test method, e.g., should the assessor carry any instruments; should any test methods be demonstrated during assessment.
5. Determine the applicability of proficiency testing for important test methods and the means for implementing new or available proficiency testing programs.
6. Recommend selection criteria and nominate technical experts who could serve as assessors and evaluators.

7. Recommend any other special technical requirements which should be considered as necessary for accreditation in this testing field.

C. *Application for Accreditation.

1. Any laboratory may request an application for accreditation in any established accreditation programs following the instructions provided in notices announcing the formal establishment of the program.

2. Upon receipt of the laboratory's application, the Director, NBS/ICST shall:

- a. Acknowledge receipt of the application;
- b. Request further information, if necessary;
- c. Confirm payment of fees before proceeding with the accreditation process; and
- d. Specify the next steps(s) in the accreditation process.

3. In accepting an application from a foreign-based laboratory, the Director, NBS/ICST shall take into consideration the policy of the host government regarding the acceptance of test data from laboratories accredited by ICST or other foreign accreditation systems.

II. TESTING METHODS

A. Commonly accepted conformance test methods.

1. Falsification Testing. Types of standards which employ this method include:

- a. Programming Language Standards
- b. Graphics Standards
- c. Text Standards

2. Proofs of Correctness Testing.

B. Guidelines for development of test method approaches. The following steps should be followed when developing a test method:

1. Management Plan. Describes the overall scheme for the testing. It should include a specification of the technical objectives to be achieved, a description of

the test strategy, and an outline of the procedures to be used for testing. This management plan should be publicly available.

2. Non-Testable Features. Although the objective is always to test all functions in a standard, there are sometimes parts of a standard which are non-testable. For example, some error conditions, depending on their nature, cannot be included in a standardized test method to check conformance. A list of all areas explicitly not tested should be available as part of the documentation of the test method.

3. Test Strategy. A full description of the test strategy to be used for the standard should be produced and made publicly available. This should include a specification of the structure of the proposed test suite, an indication of the number of tests, and an identification of the elements of the standard to be tested.

4. Test Procedures. The test method should be automated and include consideration of the procedures used to carry out testing. The setting up and execution of the test suite on a system under test, and the analysis of the output produced should be carried out in accordance with clearly defined procedures. These procedures, should be documented in a manual to accompany the test suite, and should give criteria for the analysis of all acceptable outputs for each test in the test suite.

5. Levels of Testing. The development of the test method should take into account all possible levels and/or options specified in the standard. The test method should allow for modularity so that tests for a specific level or option can be easily identified.

6. Usability. The test method should be specified in a way that is easy to understand and to implement. The resulting test suite should be useful to implementors.

C. General criteria for selecting appropriate test method.

1. Standard Specification analysis.

a. Definition of conditions for conformity with standards (conformance clauses).

b. Specification requirements. The requirements of the standard should be evaluated to identify the type of tests necessary for determining conformity.

The test method might include tests for determining that the implementation

- 1) rejects illegal structures;
- 2) accepts data and other constructs permitted by the standards, and processes them in accordance with the standard;
- 3) does not include extensions to standard;
- 4) properly flags usage of standard requirements options or levels;
- 5) properly processes implementation dependent specifications in accordance with the standard;
- 6) handles optional implementation features of the standard correctly.

2. Written test procedures which identify:

- a) test objective
- b) personnel responsible for test
- c) necessary equipment
- d) input materials, e.g., support software
- e) input data
- f) expected output
- g) constraints
- h) required test preparation, setup
- i) operator actions, console setups
- j) step-by-step execution instructions
- k) a list of any tests which are performed for information only, and will not be used to determine conformity.

3. Test suite is:

- a) easy to understand and manage;
- b) portable between different hardware

configurations and designed to take into account any implementation-dependent criteria allowed by the standard;

c) designed in such a way that there are no restrictions in using the test method or implementations on the size machine on which it can be executed.

4. Documentation Requirements.

- a. description of test cases
- b. notation used to define test
- c. Test method implementation/user guides
- d. evaluation guides for acceptable test results.

5. Requirements for specially designed test equipment, training

D. Test Suite Structure. The following guidelines should be followed when writing test suites:

1. Clear documentation should be incorporated into the tests and this documentation should include:

- a. comments, which are clear and concise. No redundant comments should be included.
- b. a reference to the clauses in the standard which are under test.
- c. clear statements of any assumptions made in the test suite design.
- d. a list of any tests which are performed for information only, and will not be used to determine conformity.

2. The test suite should be split into a series of independent test programs or scenarios, such that each test program tests only one specific requirement of the specification document.

3. Each self-contained test should use the same format, and:

- a. list all test tools necessary.
- b. provide the test environmental data, including

display of setup instructions to operator.

c. include program documentation section.

d. initialize its own workspace.

e. perform the test.

f. provide a verification section.

g. check whether passed/failed.

h. report result.

i. have the pass/fail path clearly defined.

j. include a simple comment describing the task performed by the test and a reference to the section of the standard under test.

h. be kept as simple as possible.

4. Only simple features of the programming language should be used.

5. Translating the Test Suite to Other Languages. If possible and when applicable, the test suite should be written in some common general format, which can then be converted, by the use of translators, to each of the programming languages. Updates to the test suites across languages must be kept in line at all times by strict use of the change control procedures.

E. Maintenance of Test Methods

1. Responsible Party. The Certification Body is responsible for ensuring maintenance of current test methods, authorizes changes, defines the procedures for approving changes, and indicates when any changes become effective for validation purposes.

2. Procedures. A published change control procedure should be established for both the test software and the associated documentation. The change control procedures should provide a formal mechanism for the following:

a. reporting errors in the test software.

b. logging changes to the test software.

c. reporting errors in the documentation.

d. logging changes to the documentation.

e. keeping all changes on line across all language versions.

F. Investigate Other Types of Standards Testing Methods:

1. NBS's COBOL74, BASIC and FORTRAN Compiler Validation Systems.

2. National Computer Center COBOL85 Compiler Validation Systems

3. British Standards Institution Pascal Validation System

4. GKS testing in the Federal Republic of Germany and in the UK

5. Department of Defense Ada Compiler Validation Capability

6. Defense Communications Agency, DDN compatibility testing

G. Identify any interim procedures which may be applied to all standards.

1. Program Startup Through an Established Validation Center. Since resource, equipment, and overhead costs associated with validation are relative unknowns, it may be possible to subcontract to a national or internationally recognized validating organization.

III. CERTIFICATION AND REPORTING CONSIDERATIONS

A. Certification Body.

1. General Requirements.

a. The Institute for Computer Science and Technology, National Bureau of Standards, shall serve as the Certification Body for FIPS standards.

b. Access to the Certification Body shall not be conditional upon membership in any association or group, nor shall there be undue financial conditions to restrict participation.

c. The procedures under which the Body operates

shall be administered in a non-discriminatory manner.

2. Responsibilities.

a. Approve test methods to be used by the testing laboratories when they are not included in the standard itself. Ensure the test method is technically correct before using it as the basis for issuing certificates.

b. Maintain integrity of the test method (including the test software). Note: This responsibility may be delegated to a testing laboratory.

c. Collaborate with the testing laboratory in drafting a contract for testing services to include regulations which define rights and obligations in detail.

d. Determine accreditation of testing laboratories and keep lists of all accreditations. Since technical requirements for accreditation are specific for each FIPS, technical requirements for accreditation should be developed using expert advice in that given field.

e. Notify any testing laboratory that it is no longer accredited.

f. Issue certificates at the request of the Client, based on the certification criteria defined by the Certification Body.

g. Procure, when available, an internationally recognized certificate of similar content or national certificates of other countries, at the request of the Client.

h. Define Client/testing laboratory dispute procedures. Arbitrate appeals in cases of dispute. Include procedures for disputed tests as well as disputed testing results.

i. Determine if there is a need for a Qualified Computer Products List. If there is a need (possibly on a case by case basis), identify who should maintain and what information should be reported.

j. Determine what process should be followed in

recognizing other published Certified Computer Products Lists.

k. *Notify the public to advertise new accreditation programs when the technical requirements and test methods are developed for the standard(s). Publish a notice in the FEDERAL REGISTER announcing the establishment of the program, requirements, or methods. The notice will:

1) Identify the scope of the accreditation program; and

2) Advise how to apply for accreditation.

1. Develop criteria and procedures for evaluating, modifying and accepting test methods for use in testing products for conformance to the standards.

m. Develop criteria for using test methods developed by:

1) Industry

2) Government

3) Other Countries

4) Universities

n. Publish the effective date for modifications to test methods.

o. When applicable, define the requirements and procedures of a Manufacturer's Declaration of Conformity testing program.

B. NBS/ICST Role in Conformance Testing As Government Agent

1. Assist and educate the agencies with regard to the scope and purpose of the test method as a measure of product quality.

2. Assist agencies in developing solicitation document requirements with regard to product testing for determining conformance to standards.

3. *Publicity and Announcement Sources for informing national and international industry, Government and product users on the availability of conformance test

methods and validation services. The Director, ICST, shall

a. Prepare and publish at least once each year a directory of accredited conformance testing laboratories. The Directory should contain the name and address, scope of accreditation, contact person, and the accreditation renewal date for each accredited laboratory. Distribute the directory nationally and internationally to interested manufacturers, suppliers, retailers, professional and trade associations, code groups, and government agencies.

b. Periodically prepare supplements to the directory of accredited laboratories covering new accreditation actions taken, including initial accreditations, renewals, suspensions, terminations, and revocations.

c. Make every reasonable effort to ensure the affected testing community within the scope of the specific FIPS is informed of any planned workshop. Summary minutes of each workshop will be prepared. A copy of the minutes will be made available for inspection and copying at the NBS Records Inspection Facility.

4. Incorporate requirements within the procurement regulations (i.e., General Services Administration's Federal Information Resources Management Regulations) concerning use of conformance testing and certification of products offered to the Government.

C. Role of Standards Bodies (National and International)

1. Consider conformity testing aspects in the development of standards.

2. Provide advice and give technical committee or subcommittee recommendations for the approval of a test method.

3. Provide technical advice to the Certification Body in the accreditation of a testing laboratory for performing conformity testing with respect to a specific set of standards.

4. Provide technical advice on interpretation of the standard in case of conflicts.

D. Conformity Summary Report.

1. Content.

- a. Includes a complete "test report" and an "accredited laboratory test report."
- b. Attach Client's complete computer product description for the product tested.

2. Format.

a. Accredited Laboratory Test Report Format. Will be prepared in accordance with ISO/IEC Guide 25. Please refer to the guidelines provided under V.B.7.

b. Common Test Report Format. The following guidelines should be followed when writing the code to produce the test program reports:

1) The volume of output produced by each test program should be kept as concise as possible.

2) The following details should be given in the test report for each test: test number, brief description of the test, reference to the standard, and message indicating pass/fail/ not executed.

3) Description/identification to test environment.

4) Test discrepancy summary.

3. Style for Each Test Method.

a. Each test report should provide a summary of the results giving the following information: number of tests passed, number of tests failed, number of "information only" tests, number of tests not executed, and total number of tests.

b. wherever possible, the output of test programs should be structured in a way which is easy to check both manually and automatically.

c. If test programs produce visual output on a graphics display, then this output should be kept as simple as possible to ease the checking process, while still examining all necessary

features.

d. All tests which produce visual output should be accompanied by a comprehensive check list in the form of an operator script and a set of reference samples.

e. Particular care and attention shall be paid to the arrangement of the test report, especially with regard to presentation of the test data and ease of assimilation by the reader. The format shall be carefully and specifically designed for each type of test carried out, but the headings shall be standardized as far as possible.

E. Certificate of Conformity format, content and presentation.

1. Identifies the standard(s) tested.

2. Includes a statement of the fact that the product(s) or service(s) meets the standard(s) or an applicable portion(s) thereof, clearly identifying that which is being certified.

3. Includes a clear reference to the testing laboratory and test report produced.

4. States the designation of the product version tested.

5. If any modifications to the product occur within the valid time period, the certificate only applies to the version prior to modification. All modifications must be retested for conformance.

6. Validity of the certificate depends on the set of standards, and a specified period of time.

7. Indicate laboratory accreditation by NBS/ICST.

F. Applicability of Certificate of Validation on product tested. Some of the criteria to consider:

1. Validating software on several of the system configurations for which conformance is claimed.

2. Maintenance.

3. Use of the product under different operating environments e.g., different hardware configurations and operating systems.

4. Validating across new releases of software.
5. Basing validation on documentation versus based on correct execution.
6. Basing on user experience (Better Business Bureau approach). Correct bugs which the customer complains about or lose certification. An independent Control Board is established to handle complaints.

G. Types of Certification Systems

1. Type Testing. Method under which a sample of the product is tested according to a prescribed test method in order to verify the compliance of a model with a specification. It is the simplest and most limited form of independent certification of a product both from the point of view of the manufacturer and the approval authority.

2. Type Testing followed by subsequent surveillance through audit testing of samples purchased on the open market. A system based on type testing but with some follow-up action to check that subsequent production is in conformity. Open market audit testing means a random audit testing of the type tested models from distributors' or retailers' stock.

3. Type Testing followed by subsequent surveillance through audit testing of factory samples. A system based on type testing but with some follow-up action to check that subsequent production conforms. Selected models are tested from the production prior to manufacturer's dispatch.

4. Combination of 2 and 3 above. A system based on type testing with follow-up audit testing of both factory and open market samples.

5. 100% Testing. A system under which each and every product certified is tested to the requirements of the technical specification.

6. Batch Testing. A system under which a batch of a product is sample tested and from which a verdict on the conformity with the specification is issued.

IV. TESTING ADMINISTRATION

A. General Guidelines for Test Method Administration

1. Test methods developer can be any person or institution.

2. Test methods need to be approved by the Certification Body under technical guidance of the responsible standards committee.

3. Test methods for which accreditation is offered are made available to everybody through adequate specified conditions.

4. Test methods for which accreditation is offered are official for and binding on conformity testing.

5. Same test methods derived from the standards specification should be used by all testing laboratories in all countries.

6. Standard upon which certification is based is:

a. Acceptable to buyer or governmental regulator if mandated.

b. Used in its entirety unless limitations are fully disclosed.

c. A national or international standard(s), available to the public.

d. Suitable for use as a basis for conformity.

7. Declaration of conformity is indicated by a statement, certificate, label or mark which:

a. includes information on the product, standards, producer, and procedures used.

b. present clearly the intent of the certification.

B. Criteria and procedures for Alternative Testing Methods.

1. Manufacturer's declaration of conformity (Self testing). Requirements for Vendor include:

a. Manufacturer's quality management system and documentation

b. Technically appropriate resources for testing and inspection.

c. Quality assurance based on sampling, testing,

and inspection as appropriate.

d. Retention of complaint records against product or service.

e. Any dispute resolution is in accordance with the Certification Body for that standard.

2. Third-party testing.

a. Products to be tested are fully identified.

b. Procedures by which the laboratory is operated include provision for:

1) fiscally responsible management and appropriately trained staff.

2) participation on a non-discriminatory basis.

3) both initial and continuing validation activities.

4) selection and retention of qualified testing and inspection services.

5) notification of participants on changes in standards and procedures.

6) confidentiality of proprietary information.

7) maintenance of records.

8) safeguarding use of certification mark.

9) revocation of authorization to use the certification mark.

c. Documentation required by the laboratory includes:

1) availability of published program directory listing products, standards, licensees, and other parties.

2) file of legally binding agreements for licensees.

3) availability of published statement of operating procedures.

4) issuance of annual report on system's status.

3. Government-managed testing.

C. Associated costs.

1. Costs associated with conformance testing will be on a cost-reimbursable basis. A signed Customer Agreement, along with a signed Purchase Order (or other form of negotiable money) for the total cost of the conformance testing, must be provided to the testing laboratory no later than 60 days prior to the month of any scheduled on-site data collection portion of the validation. No work will be done on the Client's product testing until funding authorization has been received.

2. Costs associated with developing and maintaining conformance test methods and responsible source of funding.

D. Resource and overhead requirements.

1. Where and what type of facilities are required to perform validations. Considerations of competence, impartiality, and integrity are fundamental to the acceptability of the testing activity. Clients of the testing services must be guided by this fact in making the choice of testing laboratories that they employ.

2. What staff, equipment and office facilities are required to support a validation service.

3. What are the retention and storage requirements for validation materials, project folders and financial accounting records.

4. How many tests will be conducted over a year.

E. Criteria and Procedures for Requesting Validation Services. The following table provides example timeframes associated with first time validation. These times will vary according to complexity of tests requested, and number of tests being performed.

SCHEDULE EXPECTATIONS FOR VALIDATION

T - 120	Contact Accredited Testing Laboratory, Negotiate agreement; vendor testing
T - 90	Submit disputed tests, if any
T - 60	Prevalidation test information/material received by the testing laboratory
T - 30	Resolve all outstanding issues
TEST	Laboratory testing
T + 4-7	Testing complete
T + 60	Final Conformity Summary Report available
***** <u>Key</u> *****	
T = Testing Day 1; '-' or '+' = Before/Beyond Test Start	

Table I.

1. Requesting validation. The Client must provide the following information/material to the Accredited Testing Laboratory in a Letter of Request:

- a. Desired and alternate month of testing
- b. Product identification, including version
- c. Host and target configurations, operating systems, program language and other appropriate information necessary for the testing laboratory to carry out the test.
- d. Site locations and addresses.
- e. A point of contact for validation, including full name, address, and telephone number.
- f. Which test(s) the Client desires to take.

2. Renewing validation. The Client must request renewal to the appropriate testing laboratory. the request must include:

a. date of current certificate of conformance.

b. a statement clearly confirming the computer product which is certified was not modified from the computer product described in the Conformity Summary Report.

3. Revalidating. When a modification has been made to the computer product previously certified beyond continuing maintenance, it is necessary for the Client to revalidate the computer product even if the certificate of conformance has not expired. The Client will follow the same procedures as IV.E.1. above. In addition the Client will observe the following:

a. identify in the computer product description the specific modifications that have been made to the product since its certification.

b. (Cross-reference)

4. Scheduling Validations. These times apply to the Client's responsibilities to the testing laboratory. The specific month in which the testing takes place is established by the laboratory.

a. Any prevalidation information/material must be received 60 days prior to the month of the scheduled validation.

b. For products not previously offered to the Government, request for validation should be submitted at least 120 days prior to the desired testing month. Data collection dates will not be established for first time validations until after a successful review of prevalidation materials has been made by the testing laboratory.

c. Requests for changes to an existing test date should be submitted no later than 60 days prior to that date.

d. Priorities. Top priority for validations is given to the scheduled annual validations. Requests from Clients for special, procurement-related validations are given second priority. Third priority is given to requests for official validation of new products. All other requests are given a lesser priority.

5. Appeal Process.

a. Disputed tests. If a Client believes any tests are in error, or not appropriate, the Client shall contact in writing the testing laboratory, and provide associated justification for the dispute. A copy of this letter shall be forwarded to the Certification Body as well. Disputed tests that are withdrawn while under dispute, could postpone testing indefinitely.

b. The Client shall follow the procedures defined by the Certification Body which accredited the testing laboratory for any appeals over disputed tests or disputed validation testing decisions by the Accredited Testing Laboratory. {Cross-reference}

V. TESTING LABORATORIES

A. Investigate National Laboratory Accreditation Programs to determine applicability to program objectives and flexibility to introduce additional specific accreditation requirements.

1. NBS National Voluntary Laboratory Accreditation Program (NVLAP). The Program purpose is to accredit laboratories for specific tests or types of tests in product or service areas where a need for accreditation has been determined. NVLAP currently has programs to accredit laboratories that test a variety of products- from concrete to telecommunications equipment. It is not funded through tax dollars, but through fees collected to cover only the costs of activities pertaining to the evaluation and accreditation of laboratories. The decision to accredit a laboratory is based on an on-site assessment by a qualified peer assessor under contract to NBS. NVLAP-accredited laboratories pay annual fees, go through on-site reassessment every two (2) years, and participate in scheduled proficiency testing to maintain accredited status.

2. American National Standards Institute Policy and Procedures & Manual of Operations for Accreditation of Certification Programs.

3. Provisions Governing Qualification (SD-6), Qualified Product List, Defense Standardization Manual, DoD 4120.3-M, "Defense Standardization and Specification Program".

4. United Kingdom NAMAS I program.

5. Australia NATLAS program.

6. New Zealand TELARC program.

B. *Conditions for Accreditation. To become accredited and maintain accreditation, a laboratory shall agree in writing to:

1. Be assessed and evaluated initially and on a periodic basis.

2. Demonstrate, on request, that it is able to perform the tests representative of those for which it is seeking accreditation.

3. Pay all relevant fees.

4. Participate in proficiency testing as required.

5. Be capable of performing the tests for which it is accredited according to the latest version of the test method within one year after its publication or within another time limit specified by the Director of ICST.

6. Limit the representation of the scope of its accreditation to only those tests or services for which accreditation is granted.

7. Limit all its test work or services for Clients to those areas where competence and capacity are available.

8. Limit advertising of its accredited status to letterheads, brochures, test reports, and professional, technical, trade or other laboratory services publications, and use the ICST logo under guidance provided by the Director, ICST.

9. Inform its Clients that the laboratory's accreditation or any of its test reports in no way constitutes or implies product certification, approval, or endorsement by NBS.

10. Maintain records of all actions taken in response to testing complaints for a minimum of one year.

11. Maintain an independent decisional relationship between itself and its Clients, affiliates, or other organizations so that the laboratory's capacity to render test reports objectively and without bias is not adversely affected.

12. Report to the Director of ICST within 30 days any major changes involving the location, ownership, management structure, authorized representative, approved signatories, or facilities of the laboratory.

13. Return to the Director, ICST, the certificate of accreditation for possible revision or other action should it:

- a. be requested by the Director of ICST,
- b. voluntarily terminate its accredited status, or
- c. become unable to conform to any of these conditions or the applicable criteria for accreditation and related technical requirements.

C. *Criteria for Accrediting Testing Laboratories. The criteria addresses a laboratory's quality system, staff, facilities, and equipment, test methods and procedures, records, and test reports. At a minimum, the following criteria must be met following ISO/IEC Guide 25, in order for a testing laboratory to be accredited:

1. Quality System

a. The laboratory shall operate an internal quality assurance program appropriate to the type, range, and volume of work performed. The quality assurance program shall be documented in a quality manual which is available for use by the laboratory staff. The quality manual shall be maintained relevant and current by a responsible member of the laboratory staff. A person or persons having responsibility for quality assurance within the laboratory shall be designated by the laboratory management and have direct access to top management.

b. The quality manual shall contain information regarding:

- 1) the structure of the laboratory (organizational charts);
- 2) The operational and functional duties and services pertaining to quality, so that each person concerned will know the extent and the limits of his responsibility;
- 3) general quality assurance procedures;

- 4) quality assurance procedures specific for each test, as appropriate;
- 5) where appropriate, proficiency testing, use of reference material, etc;
- 6) satisfactory arrangements for feedback and corrective action whenever testing discrepancies are detected;
- 7) procedure for dealing with technical complaints.

c. The quality system shall be systematically and periodically reviewed by or on behalf of management to ensure the continued effectiveness of the arrangements, and corrective action initiated. Such reviews shall be recorded together with details of any corrective action taken.

NOTE: In small laboratories, the quality system may fulfill the requirements of this clause in a simplified way.

2. Staff

a. Staff shall have the necessary education, training, technical knowledge and experience for their assigned functions.

b. There shall be a job description for each senior technical position category, which includes the necessary education, training, technical knowledge and experience.

c. The proportion of supervisory to non-supervisory staff shall be such as to ensure adequate supervision.

d. Suitable staff shall be nominated to deputize for the senior technical and quality system management staff in their absence.

e. Information on the relevant qualifications, training, and experience of the technical staff shall be maintained by the laboratory.

NOTE: In small laboratories, one person may fulfill more than one function.

3. Testing and measuring equipment

a. The testing laboratory shall be furnished with or have access to all items of equipment required for correct performance of the tests and measurements for which it is recognized.

b. All equipment shall be properly maintained to ensure protection from corrosion and other causes of deterioration. Instructions for a proper maintenance procedure for those items of equipment which require periodical maintenance shall be available.

c. Any item of the equipment which has been subjected to overloading or mishandling, or which gives suspect results, or has been shown by calibration or otherwise to be defective, shall be taken out of service and clearly labelled until it has been repaired and then shown by test or calibration to be performing its function satisfactorily.

d. Records shall be maintained on each major item of equipment. Each record shall include:

- 1) The name of the item of equipment.
- 2) The manufacturer's name and type identification and serial number.
- 3) Date received and date placed in service.
- 4) Current location, where appropriate.
- 5) Details of maintenance.

e. In the case of measuring equipment, the record shall also include:

- 1) Date of last calibration and calibration reports.
- 2) The maximum period of time between successive calibrations.

f. A label or tag indicating the date of the last calibration and the due date of the next calibration should be attached to the equipment requiring calibration.

4. Test methods and procedures

a. The testing laboratory shall have adequate documented instructions on the use and operation of all relevant equipment, on the handling and preparation of test items (where applicable), and on standard testing techniques, where the absence of such instructions could jeopardize the efficacy of the testing process. All instructions, standards, manuals, and reference data relevant to the work of the testing laboratory shall be maintained up-to-date and be readily available to the staff.

b. The testing laboratory shall use methods and procedures required by the specification against which the test items are to be tested. The specification shall be available to staff performing the test.

c. Where it is necessary to employ test methods and procedures which are non-standard, these shall be fully documented.

d. All manual calculation and data transfers shall be subject to appropriate checks.

e. Where these results are derived by electronic data processing techniques, the stability of the system shall be such that the accuracy of the results is not affected. This generally implies an ability to detect malfunctions in the hardware during program execution and take appropriate action.

5. Environment

a. The environment in which the tests are undertaken shall not invalidate the test results or adversely affect the required accuracy of measurement. The testing premises shall be protected as required from excessive conditions such as excessive temperature, dust, moisture, steam, vibration, electromagnetic disturbance, interference, and shall be maintained accordingly. They shall be sufficiently spacious to limit the risk of damage or danger and to allow operators to make practical and precise movements. The premises shall have the equipment and energy sources needed for the testing. When the testing so requires, they shall be equipped with devices to monitor the environmental conditions.

b. Access to and use of all test areas shall be controlled in a manner appropriate to their designated

purpose and entry by persons external to the laboratory shall be defined.

c. Adequate measures shall be taken to ensure good housekeeping in the testing laboratory.

6. Records

a. The testing laboratory shall maintain a record system to suit its particular circumstances and comply with any existing regulations. It shall retain on record all original observations, calculations and derived data, calibration records, and the final test report for an appropriate period. The records for each test must contain sufficient information to permit satisfactory repetition of the test.

NOTE: In some countries it may be necessary to maintain records for a period specified by law.

b. All records and test reports shall be held secure and in confidence to the Client, unless otherwise required by law.

7. Test reports

a. The work carried out by the testing laboratory shall be covered by a report which accurately, clearly, and unambiguously presents the test results and all other relevant information.

b. Each test record shall include at least the following information:

- 1) name and address of testing laboratory;
- 2) unique identification of report (such as serial number), and of each page of the report;
- 3) name and address of Client;
- 4) description and identification of the test item;
- 5) date of receipt of test item and date(s) of performance of test, as appropriate;
- 6) a statement to the effect that the test results relate only to the items tested;
- 7) identification of the test specification, method, and procedure;

- 8) description of sampling procedure, where relevant;
- 9) any deviations, additions to, or exclusions from the test specification, and any other information relevant to a specific test;
- 10) disclosure of any non-standard test method or procedure utilized;
- 11) measurements, examinations and derived results, supported by tables, graphs, sketches, and photographs as appropriate, and any failures identified;
- 12) a statement on measurement uncertainty (where relevant);
- 13) A signature and title of person(s) accepting technical responsibility for the test report and date of issue;
- 14) a statement that the report shall not be reproduced except in full without the approval of the testing laboratory.

c. Corrections or additions to a test report after issue shall be made only by a further document suitably marked, e.g., "Supplement to test report serial number . . . (or as otherwise identified," and shall meet the relevant requirements of the preceding paragraphs.

8. Role of Certification Body in Accrediting.

- a. Ensure the testing laboratory is accessible to anyone, even foreign Clients.
- b. Develop a model contract (e.g., order form) to serve as agreement between Client and testing laboratory.
- c. Ensure the amount of accredited laboratories is consistent with the Client needs, without over saturating the market. Any Client requesting testing for a given standard, should be able to have his product tested within one year after his letter of request to an Accredited Testing Laboratory.
- d. Establish specific criteria for assessing competence of laboratories to perform test methods. On-site

assessment can be carried out by the Certification Body or by independent assessors approved by the Certification Body.

e. Construct procedures with enough flexibility to accommodate technological developments.

f. *Select any/all technical experts, assessors and evaluators necessary for accrediting a testing laboratory.

g. *Prepare an administrative recommendation that the laboratory either be granted or denied accreditation. This recommendation is based on a review of the evaluation and other records to ensure that all ICST technical, financial, and administrative obligations have been satisfied.

D. *Testing Laboratory Assessment. Figure 1 is a pictorial explanation of the following procedures for testing laboratory assessment:

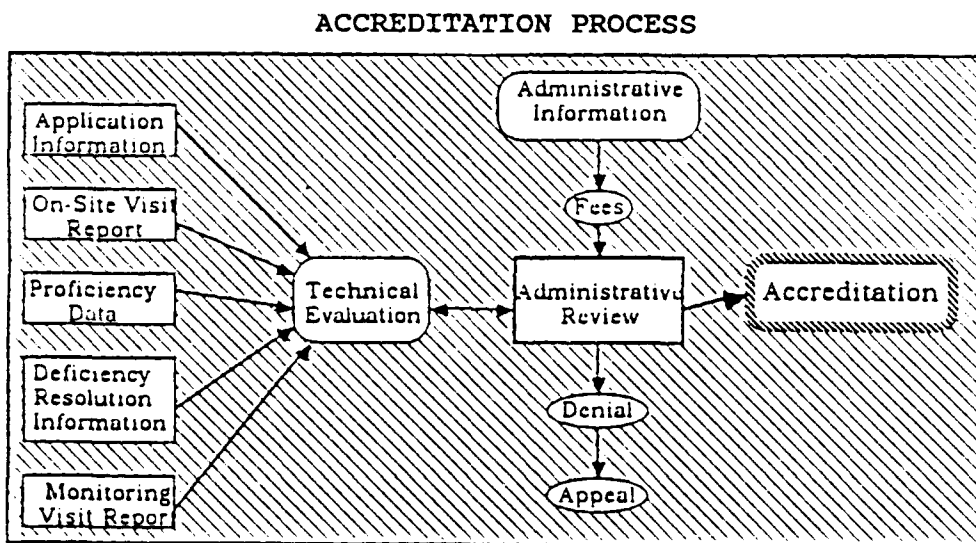


Figure 1

1. On-Site Assessment. Before initial accreditation and about every two years thereafter, an on-site assessment of each laboratory is conducted to determine compliance with the criteria. Assessors use standardized checklists so each laboratory receives a fair assessment in relation to others; however, assessors have considerable latitude in judgments about each laboratory's compliance with the criteria depending on their experience and the unique circumstances of each laboratory. The assessors are

selected and assigned on the basis of their expertise in the testing techniques to be reviewed. The time needed to conduct an assessment varies, but generally two days is the norm. Every effort is made to conduct an assessment with as little disruption as possible to the normal operations of the laboratory. The assessors:

- a. Meet with management and supervisory personnel responsible for the laboratory's activities for which accreditation is being sought to acquaint the individuals involved and to set the assessment agenda.
- b. Examine the quality system employed by the laboratory. The history of one or more samples from receipt to final issuance of test reports is traced. Assessors thoroughly review the laboratory's quality manual or equivalent, examine technician notebooks for records pertaining to the samples, check sample identification and tracking procedures, determine whether the appropriate conditions are maintained, and examine copies of completed test reports.
- c. Review records of periodic internal audits, use of check samples or participation in round robin testing or other similar programs.
- d. Review representative records including competency evaluations for all staff members who perform the tests, verification records, and sample control records.
- e. Observe demonstrations of testing techniques and discuss them with the technical personnel to assure their understanding of the procedures.
- f. Examine major equipment, apparatus, and facilities.

At the conclusion of the assessment, an exit briefing is held to discuss assessment findings with laboratory management and identify any deficiencies uncovered. A written summary of all identified deficiencies is left at the laboratory. Assessment forms and a written report is submitted to NBS/ICST for further evaluation. The laboratory is asked to respond within 30 days of the date of the exit briefing and provide documentation or certification that the specific deficiencies have been corrected or that specific actions are being taken. Any laboratory applying for initial accreditation may request a delay in responding.

If any deficiencies are noted at laboratories which are currently accredited, such deficiencies must be corrected

within 30 days after the exit briefing or the laboratory may face possible suspension, revocation or expiration of its accreditation. When test systems are identified as malfunctioning, it must not be used until corrective action has been completed. Any deficiencies noted for corrective action will be subject to thorough review and verification during subsequent assessments.

2. Monitoring Visits. In addition to regularly scheduled assessments, monitoring visits can be made at any time during the accreditation period. Monitoring visits serve to verify reported changes in the laboratory's personnel, facilities, and operations, or to explore possible reasons for poor performance in proficiency testing. The scope of a monitoring visit may range from checking a few designated items to a complete review. Failure to cooperate with NBS/ICST assessors may be grounds for adverse accreditation action. No additional fee is required for the monitoring visit since cost is already factored into the fees.

3. Proficiency Testing. Proficiency testing is an integral part of the NVLAP accreditation process. While the existence of facilities, equipment, and personnel which satisfy the criteria indicates a laboratory's overall capability to obtain good results, an analysis of actual test results for certain test methods is also necessary to determine if the overall capability does in fact produce the desired results. A laboratory's failure to participate fully in the conduct of required proficiency testing is grounds for adverse accreditation action.

4. Evaluation. Evaluation of a laboratory is conducted at NBS by technical experts who review records on each applicant laboratory and base their evaluation on:

- a. Information provided on the application;
- b. On-site assessment reports;
- c. Actions taken by the laboratory to correct deficiencies;
- d. Results of proficiency testing; and
- e. Information from any monitoring visits of the laboratory.

If the technical evaluation reveals additional deficiencies, written notification describing them will be made to the laboratory. The laboratory must respond within 30 days of such notification and provide documentation or certification that the specified deficiencies have been

corrected. Clarification of some issues may be requested by telephone. All deficiencies must be corrected before accreditation can be granted or renewed.

5. Accreditation Actions. The Director, NBS/ICST, has authority to make any of the following accreditation decisions:

a. Recommended. When accreditation is recommended, the recommendation forms the basis for granting accreditation. A certificate of accreditation is issued to the laboratory. An NBS/ICST boilerplate Certificate of Accreditation is included as Appendix E.

b. Denial. In cases where denial is recommended, the laboratory is notified of a proposal to deny accreditation and the reasons for denial.

c. Appeal. When denial has been proposed, the laboratory may request a hearing, under United States Code (U.S.C.) 556, within 30 days of the date of receipt of the notification. If a hearing is not requested, the denial becomes final upon the expiration of that 30 day period.

d. Renewal. Accreditation is granted annually or biennially with renewal occurring on the same anniversary date every year or every two years.

e. Termination. A laboratory may voluntarily terminate its accreditation by written request at any time. The accreditation certificate must be returned with the request. If a laboratory elects not to renew its accreditation, a notification of such intention should be forwarded to NBS in writing.

f. Suspension. If an accredited laboratory develops problems or deficiencies which are of a temporary nature, its accreditation may be suspended until such time as the deficiencies are resolved.

g. Revocation. In cases where a laboratory is found to have violated the terms of its accreditation, the accreditation can be revoked. The laboratory may, however, be given the option to voluntarily terminate accreditation. The laboratory has 30 days from the date of receipt of a notice of proposed revocation in which it may appeal the proposed revocation by requesting a hearing. If a hearing is not requested, the revocation becomes final upon the expiration of that 30 day period. When revocation is final the laboratory must return its certificate of accreditation

and cease to reference its NBS/ICST accreditation on any of its reports, other correspondence, or advertising.

6. Public Notification. Accreditation actions are published quarterly. A directory of accredited laboratories is published annually.

E. Responsibilities of Testing Laboratories.

1. Carry out conformity testing and provide test reports to the Client.

2. Fulfill contractual commitment to the Certification Body.

3. Record technical test details.

4. Communicate the approved test method to the Client for conformity testing preparation.

5. Treat all test results and documents confidentially, except those which are explicitly stated as public:

a. For Ada, Minimal BASIC, COBOL and FORTRAN, the ICST in the United States will provide test reports to anyone upon request. (Under the Freedom of Information Act.)

b. European testing laboratories having agreements with ICST for COBOL and FORTRAN require this right in the validation contract, restricted to the case where a certificate is issued.

c. For Pascal, the British Standards Institution agreed with industry on specific public release text in the validation contract.

6. Comply with all requirements for accreditation.

7. *Comply with existing laws. Accreditation does not relieve the laboratory of the need to observe and comply with existing Federal, State, and local statutes, ordinances, or regulations that may be applicable to its operation, including consumer protection and antitrust laws.

8. *Accredited laboratories are encouraged, within specified limits, to publicize their accredited status. The major restriction is that advertising must not imply product certification by NBS or the U.S. Government. Laboratories and their Clients may not reference their

accredited status in consumer media, in product advertising, or on product labels, containers, or packaging.

VI. MUTUAL RECOGNITION OF VALIDATION/CERTIFICATION

A. Mutual recognition of test reports means issuing a national certificate solely on the grounds of a test report, without testing anew. The layout of both test reports and certificates should be harmonized to improve mutuality among testing laboratories and certification bodies, and to ease translation.

B. Levels and organizations/committees (both national and international) for conformance testing program interface to achieve common recognition of test methods.

1. The certification system may provide the right for every testing laboratory to attend the testing by any other laboratory once a year in order to allow reciprocal observation for the sake of identical results.

2. Any Client worldwide, shall be able to address national certification bodies.

C. Considerations and issues to be resolved in order to achieve common recognition of validation certificates for products offered by multi-national companies who trade with other countries.

1. Bilateral agreements between the U.S. and Australia, New Zealand, and Great Britain, the laboratories, accredited by their respective accreditation systems: NVLAP, NATA, TELARC, and NAMAS are recognized by the parties to each agreement. The bilateral agreements were reached after mutual review of each party's system to assure equivalency of procedures and practices that lead to a laboratory's accreditation.

2. Examine bilateral agreements on mutual recognition of certificates which exist for compilers for several programming languages.

VII. CONFORMANCE TESTING REQUIREMENTS FOR PROGRAM APPLICATIONS

A. Establish procedures for testing specific applications for various governmental programs.

1. The mechanisms employed by the test suite to update implementation-dependent variables should be simple to implement and well documented.

2. Test suites for graphics systems which require a

language binding for use by the application, should be developed on the certified compiler which conforms to the corresponding language standard using no extensions.

VIII. LEGAL ISSUES

A. Legal and copyright issues:

1. Breach of proprietary software.
2. Limitations of the testing, e.g., a product may conform to the specification, but the integrity of the data transferred by that product is not certified.
3. Who owns the test suite(s) and data rights.
4. With follow-on acquisitions added to existing system(s), who is responsible for the integration/linking of various versions of certified products.
5. The validity of the certificate of conformance in light of user's complaints.
6. Interpretation of the standard for test coding.
7. Liability of the testing laboratory for subsequent failure of the product.

IX. REFERENCES

- A. "NBS Software Standards Testing Program," NBS/ICST, June, 1986.
- B. "Open Systems in Software," NBS Research Reports, September, 1985.
- C. Wood, Helen, "Emerging Software Standards: Opportunity and Challenge," Proceedings of Computer Standards Conference-1986, IEEE Computer Society Press, 1986.
- D. Kruckeberg, F., "Conformity Testing and Certification for Information Processing Systems," Computers & Standards 4, North-Holland, pp 21-32, 1985.
- E. ISO Guide 2, 1983-07-01
- F. ISO/IEC Guide 38, 1983-12-01
- G. "Certification - Principles and Practice," ISO, Geneva, 1980.
- H. Berger, Harvey (X4016), The National Voluntary Laboratory

Accreditation Program, "American Laboratory," Nov. 1986.

I. ISO/TC 97/SC 22/WG 12 N45 - "Test Methods for Programming Languages Processors - Guidelines for their development and procedures for their approval"

J. Evaluation Report on Version 1.1 of the GKS Test Software- Paul Stratton, NCC.

K. Validation Guide, LSAR ADP Systems, USAMC Materiel Readiness Support Activity, Lexington, Kentucky 40511-5101, January 1987.

L. ISO/IEC Guide 40, 1983-12-01

M. "The American National Standards Institute Policy and Procedures & Manual of Operations for Accreditation of Certification Programs," (undated)

N. Institute for Computer Sciences and Technology Compiler Testing Procedures, NBS/ICST-87/010, February 1, 1987.

O. ISO/IEC Guide 25, 1982-12-15

P. NBS Special Publication 500-98, Planning for Software Validation, Verification, and Testing

Q. NBS Special Publication 500-136, An Overview of Computer Software Acceptance Testing

R. ISO/IEC Guide 28, 1982-12-15

Potential Follow-Up References

- ISO/IEC Guide 7, 1982-12-15, Requirements for standards suitable for product certification

- ISO/IEC Guide 12, 1977-02-15, Comparative testing of consumer products

- ISO/IEC Guide 16, 1978-03-01, Code of principles on third party certification systems and related standards

- ISO/IEC Guide 22, 1982-04-15, Information on manufacturer's declaration of conformity with standards or their technical specifications

- ISO/IEC Guide 23, 1982-03-01, Methods of indicating conformity with standards for third-party certification systems

- ISO/IEC Guide 30, 1981-08-01, Terms and definitions used in connection with reference materials

- ISO/IEC Guide 31, 1981-09-15, Contents of certificates of reference materials
- ISO/IEC Guide 39, 1983-12-01, General requirements for the acceptance of inspection bodies
- ISO/IEC Guide 42, 1984, Guidelines for a step-by-step approach to an international certification system
- ISO/IEC Guide 43, 1984, Development and operation of laboratory proficiency testing

NBS PUBLICATIONS ON SOFTWARE STANDARDS

FEDERAL INFORMATION PROCESSING STANDARDS

Federal Information Processing Standards Publications (FIPS PUBS) are developed by NBS and issued under the provisions of the Federal Property and Administrative Services Act of 1949, as amended; Public Law 89-306 (79 Stat. 1127); Executive Order 11717 (38 FR 12315); and Part 6 of Title 15 of the Code of Federal Regulations (CFR).

FIPS PUBS include standards, guidelines, and program information documents for computer software, hardware, data and operations. A complete list of FIPS is available from:

Standards Processing Coordinator (ADP)
National Bureau of Standards
Institute for Computer Sciences and Technology
Technology Building, Room B64
Gaithersburg, MD 20899
Telephone: (301) 975-2816

FIPS PUBS are sold by:

National Technical Information Service (NTIS)
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Telephone: (703) 487-4650

FIPS have been issued for major programming languages, including: COBOL, Fortran, Basic, Pascal, Ada (Ada is a trademark of the U.S. Government, Ada Joint Program Office), and MUMPS. A FIPS for C has been proposed.

Other FIPS have been approved for GKS, Database SQL, Database NDL, CGM and DDF. Future FIPS are planned for data models, data definition formats, data dictionary software, communication protocols, and additional computer graphics software, in addition to a future FIPS proposal for POSIX.

SPECIAL PUBLICATIONS AND OTHER REPORTS

These publications present the results of NBS studies and research. Stock numbers and July 1986 prices are indicated. They are available from one of the following:

Superintendent of Documents
U.S. Government Printing Office
Washington, DC 20402
Telephone: (202) 783-3238

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: (703) 487-4650

The following documents cover data management, and programming languages:

NBS SP 500-118	A Guide to Performance Evaluation of Database Systems SN 003-003-02624-7	\$2.25
NBS SP 500-117	Selection and Use of General-Purpose Programming Languages Vol 1 - Overview SN 003-003-02612-3	\$3.00
	Vol 2 - Program Examples SN 003-003-02613-1	\$5.50
NBSIR 85-3164	A Technical Overview of the Information Resource Dictionary System PB 85-224483	\$16.95
NBSIR 85-3165	Using the Information Resource Dictionary System Command Language PB 85-227783	\$11.95
NBS SP 500-115	Report on Approaches to Database Translation SN 003-003-02583-6	\$3.25
NBS SP 500-108	Guide on Data Models in the Selection and Use of Database Management Systems SN 003-003-02543-7	\$3.00
NBS SP 500-131	Guide for Selecting Microcomputer Data Management Software SN 003-003-02682-4	\$2.50
NBS SP 500-132	Benchmark Analysis of Database Architecture: A Case Study SN 003-003-02684-1	\$7.50

NBSIR 85-3173	Reference Model for DBMS Standardization PB 85-225217	\$11.95
NBSIR 86-3324	Data Administration Workshop Proceedings PB 86-191152	\$22.95
NBS SP 500-122	Guide on Logical Database Design SN 003-003-02631-0	\$4.50
NBS SP 500-127	Workshop on Analytical and Simulation Modeling of IEEE 802.4 Token Bus Local Area Networks SN 003-003-02660-3	\$9.50
NBS SP 500-128	Starting and Operating a Microcomputer Support Center SN 003-003-02683-2	\$1.75
NBS SP 500-129	Software Maintenance Management SN 003-003-02681-6	\$2.75
NBS SP 500-130	Executive Guide to Software Maintenance SN 003-003-02685-9	\$1.00
NBS SP 500-131	Guide for Selecting Microcomputer Data Management Software SN 003-003-02682-4	\$2.50
NBS SP 500-132	Benchmark Analysis of Database Architectures: A Case Study SN 003-003-02684-1	\$7.50
NBS SP 500-133	Technology Assessment: Methods for Measuring the Level of Computer Security SN 003-003-02686-7	\$8.00
NBS SP 500-134	Guide on Selecting ADP Backup Processing Alternatives SN 003-003-02701-4	\$1.75
NBS SP 500-135	Integrated Software for Microcomputer Systems SN 003-003-02711-1	\$1.75
NBS SP 500-136	An Overview of Acceptance Testing of Computer Software SN 003-003-02712-0	\$1.00
NBS SP 500-137	Security for Dial-Up Lines SN 003-003-02723-5	\$3.75
NBS SP 500-138	A Functional Model for Fourth Generation Languages	

	SN 003-003-02731-6	\$2.25
NBS SP 500-139	Data Base Directions Information Resource Management --- Making It Work	
	SN 003-003-02738-3	\$9.00
NBS SP 500-140	Personal Computer Networks	
	Sn 003-003-02746-4	\$3.25
NBS SP 500-141	Annotated Bibliography on Software Maintenance	
	SN 003-003-02756-1	\$6.50
NBS SP 500-142	A Management Overview of Software Reuse	
	SN 003-003-02757-0	\$1.50
NBS SP 500-143	Guide to the Selection and Use of Fourth Generation Languages	
	SN 003-003-02758-8	\$3.25
NBS SP 500-144	Guidance on Software Package Selection	
	SN 003-003-02773-1	\$6.00
NBS SP 500-145	Programming Languages for Knowledge-Based Systems	
	SN 003-003-02783-9	\$4.00
NBS SP 500-146	Report on the NBS Software Acceptance Test Workshop	
	SN 003-003-02793-6	\$2.75
NBS SP 500-147	Guidance on Requirements Analysis for Office Automation Systems (Update)	
	SN 003-003-02791-0	\$5.50

A list of NBS publications dealing with these and other subject areas is available from:

National Bureau of Standards
 Institute for Computer Sciences and Technology
 Technology Building, Room B151
 Gaithersburg, MD 20899
 Telephone: (301) 975-2832

SN numbers - stocked by GPO
 PB numbers - stocked by NTIS

CONFORMANCE TESTING STRATEGY FOR STANDARDS OF INTEREST TO CALS

INTRODUCTION

The purpose of this Plan is to provide a generic framework under which all conformance testing programs can be developed. This appendix is provided in order to retain this generic approach in the body of the document, yet provide specific attention to such governmental programs as the Computer-Aided Acquisition and Logistics Program.

The following sections are included:

Conformance Testing Strategy. This section identifies those standards NBS currently believes of interest to CALS. It is done as a question and answer survey, and provides a snapshot status of conformance testing programs given the current scheduled resources. Status updates in future fiscal year deliverables will assume this information as background.

Standards Having CALS-Specific Testing Requirements. In some cases, national and international conformance testing programs will not be specific enough to address CALS requirements or CALS tailoring of standards. This section identifies those standards and the current status of their testing programs specific to CALS.

Testing Site Candidates. Although easier to identify testing sites by specific standard and interested party, this section provides a list of potential candidates for OASD (P&L) to consider during future budgeting cycles and resource allocations.

Industry-Funded Cooperative Arrangement. The industry has been contributing actively in the development and review of key CALS documents. This section suggests some potential areas for industry to contribute talent and time, in support of conformance testing.

CONFORMANCE TESTING STRATEGY

A. Product Data

1. **WHAT ARE THE STANDARDS?** IGES, Version 1.0-1980, Version 2.0-1983, and Version 3.0-1986, Version 4.0 (anticipated 8/87)

WHAT IS THE CURRENT STATUS OF TESTING?

Versions 2.0 and 3.0: Individual users are each performing their own testing program on translators to and from the IGES format. Test cases are available from NBS. A file analyzer is available from two commercial sources, one domestic, one foreign.

Version 4.0: Testing methodology is being developed. Existing test cases are being documented, new test cases are being created, and software tools are being developed.

WHO'S DEVELOPING THE TESTS?

Test cases are being developed by NBS, and edited by the IGES Organization, by CAD System vendors, and by user companies.

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

Yes. CALS has defined three applications subsets of IGES that will require validation testing and acceptance testing.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

A testing program for IGES would involve a rigorous methodology, a comprehensive suite of test cases, documented criteria for measuring the degree of success, and various software tools to assist in the process. This program is being created in 1987.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

NBS is committed to providing a neutral digital format for product data exchange. Through its IGES Technical Committees, a testing program will be initiated by a third party organization with oversight being provided by NBS and its IGES Organization.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

Initial operation by September, 1987.

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

Testing is being done by major users of IGES. Testing is also being coordinated by the IGES Testing Methodology Committees under a Project Manager reporting to NBS. Industry left to themselves have been unable to develop

production, quality end to end data exchange capability. The Society of Automotive Engineers have noted their willingness and their commitment to digital data exchange.

INTERIM SOLUTIONS?

Develop guidelines for users to validate their own IGES translators. Publicize the experiences of sophisticated users. Make available good quality test cases.

ASSOCIATED COSTS?

Optimum program would be \$2.5-3M/Year with some "calibration" income from industry reimbursables.

2. WHAT ARE THE STANDARDS? VHDL

WHAT IS THE CURRENT STATUS OF TESTING?

Currently, possible validation and verification developers, test suites, and administration of conformance testing is being considered.

WHO'S DEVELOPING THE TESTS?

United Technologies Microelectronics Center is currently performing verification and validation of the VHDL and support environment. This organization as well as, Intermetrics and CAD Language Systems are possible candidates.

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

Probably. TBD after IEEE standardization of VHDL and refinement of the CALS electronics standards information model.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

TBD

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

TBD; however, any organization is a possibility as long as it is accredited by NBS or other recognized accrediting program.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

After forthcoming IEEE standardization of VHDL this Fall.

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

TBD

INTERIM SOLUTIONS?

TBD

ASSOCIATED COSTS?

TBD

3. WHAT ARE THE STANDARDS? EDIF

WHAT IS THE CURRENT STATUS OF TESTING?

Nothing at this time. Second version just recently became available.

WHO'S DEVELOPING THE TESTS?

TBD

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

Possibly, needs to be integrated with VHDL, which needs interfacing with CALS.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

TBD

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

TBD

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

TBD

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

EDIF Committee.

INTERIM SOLUTIONS?

TBD

ASSOCIATED COSTS?

TBD

B. Graphics

1. WHAT ARE THE STANDARDS? CGM, GKS, GKS-3D, PHIGS, CGI

WHAT IS THE CURRENT STATUS OF TESTING?

CGM - The Europeans have offered to begin testing. NBS will spec out a reference implementation for more detailed testing.

GKS - Tests are in place. The Europeans will begin testing in June, 1987. NBS will begin testing shortly thereafter.

GKS-3D - The Europeans are just beginning tests.

PHIGS, CGI - NBS is attempting to build interest within the vendor/user community.

WHO'S DEVELOPING THE TESTS?

CGM - The European community is developing some of the tests, but NBS is doing most of the development.

GKS - Tests are being developed by the European community with NBS assistance.

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

Need identified only for CGM at this time.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

Need a software test suite for each standard with different levels of testing: internal tests and visual operator tests. Need to watch the tests for interpretation of visual output.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

CGM - Either NBS will perform the tests for FIPS CGM, a third party, or a manufacturer's declaration of conformity will be performed by commercial software developers.

GKS - NBS will perform the tests for FIPS GKS. NCC will act as the NBS agent for GKS.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

CGM - 3 years

GKS - 1 year

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

CGM - Possibly some trade association, e.g., SIGGRAPH or NCGA.

GKS - NBS initially; eventually perhaps a trade association, e.g., SIGGRAPH or NCGA.

INTERIM SOLUTIONS?

CGM - Develop a small subset and let vendors perform manufacturer's declaration of conformity.

GKS - Use NCC as the agent to do testing for NBS.

ASSOCIATED COSTS?

CGM - \$1-3M

GKS - \$100-400K to get everything in place.

C. Text

1. WHAT ARE THE STANDARDS? SGML (ISO 8879)

WHAT IS THE CURRENT STATUS OF TESTING?

NBS is developing an SGML validation suite which is to be used to validate SGML parsers. The test suite is expected to be completed by August 1987.

The ideal goal would be to state, that after successful completion of the test suite, a software product is free from errors. This will not be possible in the general case and the best we can do is give the user confidence that the product is likely to perform as described.

WHO'S DEVELOPING THE TESTS?

NBS is developing the tests in conjunction with developers of SGML parsers, i.e., through an iterative process of validating existing parsers we are improving our own validation suite.

NEED FOR SUPPLEMENTAL TESTS FOR CALS? Yes.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

For SGML - Generally speaking, a parser is a program used to determine the underlying structure and content of some input object (file, document, etc.) More formally (in an SGML context), a parser checks that the tokens appearing in the input document occur in patterns that are permitted (by the rules of SGML and the description given by the document architect in the document type definition) and makes explicit the hierarchical structure of the incoming token stream by identifying which parts should be grouped together.

Standard test suites should be developed for SGML parsers to be used by developers, users, or third-party testers. Test suites should be considered as evolving rather than static as they will be updated and expanded based on users' experience with parsers. The existence and acceptance of these test suites should lead to comparability and wide acceptance of test results produced by different examiners.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

NBS recommends manufacturer's declaration of conformity using the validation suite to be defined by ICST (in conjunction with other implementors).

The main focus of SGML conformance testing will center on 'live' processing of test document(s) by the parser being tested, i.e., documents - conforming and/or non-conforming, are given to the parser and a verdict is determined for its behavior. If the document is conforming, the parser should report no errors; if the document is non-conforming, the parser should note an

error. An observer will analyze the outcome and, if it is as expected, the observer will consider the test successfully completed, else he will consider it failed.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

The validation suite will be available in the summer of 1987. NBS recommends manufacturer's declaration of conformity, and so plan to distribute the validation suite to requestors.

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

Manufacturer's declaration of conformity. NBS tests will be developed in concert with implementors of SGML parsers.

INTERIM SOLUTIONS?

The validation suite is being developed in a phased and recursive manner. Until it is complete, SGML software developers are "testing" the test suite.

ASSOCIATED COSTS?

The cost of developing the test suite will be paid by CALS; costs of running the test suite would be placed on the software developer.

OTHER CONSIDERATIONS.

The testing of SGML parsers presents a particularly challenging situation because:

1. Parser output is loosely defined in the SGML standard; also, the parser will be incapable of helping to diagnose its own mistakes (as contrasted to a compiler or interpreter which could perform some process and compare the outcome - in some cases - with a constant known value.)
2. Only minimal output is required by the parser - the parser is required only to report whether or not an error was encountered; no standard reporting form is required. Therefore, much of our evaluation of a parser's correct or incorrect handling of some function will be by inference, e.g., we cannot know that a parser has correctly interpreted an attribute value but we will infer that it has properly recognized the attribute value if it reports no error for a correct value and does report an error for incorrect values.
3. The tests cannot be modeled from the parser design because that will be unknown to the persons conducting the tests.

4. Various levels of implementation are likely since there are several functions in the standard which may not be useful to most users. The tests should be structured so that failure to process some rarely used function of the language will not disqualify a parser from further evaluation.

5. There is no requirement that an SGML parser continue after encountering an error, therefore, the number of exception test documents will be relatively large.

6. There are parts of the standard for which validation may not be possible, e.g., a validating parser which is not associated with any sort of formatting output process may fail to recognize 'record ends' which are caused by markup.

7. Finally, as with any complex computer application, complete validation is a goal that may never be attained. A failed test shows that a parser implementation does not conform to the standard; a successfully completed test shows only that it may conform. The completion of a series of well constructed tests establishes confidence that the software will perform as intended.

2. WHAT ARE THE STANDARDS? ODA/ODIF

WHAT IS THE CURRENT STATUS OF TESTING?

No testing at this time; however, NBS intends to develop test documents and other tests for the interchange format (ODIF) as soon as an ODA implementation is available. NBS is attempting to build interest in the vendor/user community for the development of implementations and tests.

WHO'S DEVELOPING THE TESTS?

NBS intends to develop the tests.

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

Possibly. It is dependent on whether OSD wants compliance with specific military document types versus just conformance with the ODA standard.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

As soon as an ODA implementation is developed for testing, requirements would be similar to those defined for SGML.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

TBD

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

1-3 Years.

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

TBD

INTERIM SOLUTIONS?

TBD

ASSOCIATED COSTS?

TBD

D. Data Management

1. **WHAT ARE THE STANDARDS?** Database Language SQL, Database Language NDL

WHAT IS THE CURRENT STATUS OF TESTING?

NBS is attempting to build interest in the vendor/user community for the development of tests. Preliminary goals include: a separate suite requirement for each host language binding; SQL has the highest priority.

WHO'S DEVELOPING THE TESTS?

No organization has agreed to take the lead at this time.

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

No need identified at this time.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

Determine if an implementation conforms to both the FIPS and ANSI standards. A test suite should determine conformance to Level II (includes Level I) of each ANSI standard.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

Onsite "witnessed" testing as is currently used for other programming languages, may be required.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

2-4 Years

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

TBD

INTERIM SOLUTIONS?

Users should purchase software only from implementors who guarantee conformance, i.e., the vendor will modify their product whenever errors are discovered.

ASSOCIATED COSTS?

Three people fulltime for at least 2 years, or an equivalent contracted effort.

2. WHAT ARE THE STANDARDS? IRDS

WHAT IS THE CURRENT STATUS OF TESTING?

Two actions are currently underway:

- NBS has a contract with AOG Systems Corporation for new IRDS specifications development. Within the contract there is a tasking to develop techniques for determining whether commercial dictionary software products are in conformance with the specifications of the IRDS. This is scheduled to be completed August, 1987.

- Through the NBS Research Associate Program, NBS has a pending Memorandum of Agreement (MOA) with a major data dictionary software vendor to develop techniques for determining whether commercial dictionary software products are in conformance with the IRDS. No completion date has been determined.

WHO'S DEVELOPING THE TESTS?

NBS with assistance from AOG Systems Corporation and a major data dictionary software vendor (through the Research Associate Program).

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

Possibly for a data dictionary/directory system of CALS-related data elements.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

- Decompose the IRDS into a set of logical components.
- Define the nature of the standard data to be applied to testing.
- Demonstrate how tests would handle output from multiple IRDS implementations.
- Develop an abstract, hierarchically structured model of the IRDS with components that can be separately tested.
- Determine conformance criteria for individual components of the IRDS model.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

Automated tests for the IRDS components where feasible.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

1-2 Years for selected components.

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

NBS goal is a third party testing laboratory, although it may be necessary for NBS to provide testing on an interim

basis.

INTERIM SOLUTIONS?

A reference implementation developed by NBS could be used as a basis for testing criteria.

ASSOCIATED COSTS?

Funding may be required to accelerate development of needed tests. The amount is currently undetermined.

3. WHAT ARE THE STANDARDS? DDF, ASN.1

WHAT IS THE CURRENT STATUS OF TESTING?

No plans as of this date. NBS is using ASN.1 extensively in its OSI File Transfer Access Mechanism (FTAM) initiative. A conformance testing approach may follow from this work.

WHO'S DEVELOPING THE TESTS?

TBD

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

TBD

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

TBD

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

TBD

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

TBD

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

TBD

INTERIM SOLUTIONS?

TBD

ASSOCIATED COSTS?

TBD

E. Communications.

1. WHAT ARE THE STANDARDS? Government Open Systems Interconnection Procurement (GOSIP) Specification

WHAT IS THE CURRENT STATUS OF TESTING?

Cooperative testing among vendors through participation in demonstrations, OSINET operation, and research and development activities at NBS are all initiatives

contributing to the testing program development.

WHO'S DEVELOPING THE TESTS?

Some tests have been developed by NBS. The Corporation for Open Systems (COS) is developing tests. The Industrial Technology Institute (ITI) already offers tests. Tests will be evaluated by NBS for their suitability/sufficiency in meeting Federal requirements.

NEED FOR SUPPLEMENTAL TESTS FOR CALS?

There may be a potential need, but it's not clear at this time.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

Government agencies procuring OSI conforming networks may require testing for conformance to standards. The tests to be administered and the testing authority are at the discretion of the agency Acquisition Authority. Some possible requirements may be the development of a test architecture, test systems, test languages and tests. In addition, an accreditation process needs to be developed if "manufacturer's declaration of conformity" is the administering methodology.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

Conformance tests of the functional units comprising the layered architecture may be administered by a recognized testing service or may be self-administered by the vendor. Currently, there is no recognized testing service. Multi-layer testing or protocol suite testing, or a packaged product may be used by the vendor, user, or third party, at the discretion of the agency Acquisition Authority.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

Since the OSI specification is new, it will take some time for testing services to become established; however, some organizations are already showing an interest by preparing to offer services, and some test capability should be available by the summer of 1988.

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

A testing service or manufacturer's declaration of conformity under an accreditation process seem to be the most desirable, and the most likely methods to be supported by industry. The manufacturer will most likely perform single layer testing with the testing service providing suite testing when available. Possible candidates for a testing service include: Corporation for Open Systems, National Computer Center in UK, European manufacturers, the Japanese Consortia, or the OSI Lab at

the Washington Navy Yard.

INTERIM SOLUTIONS?

NBS coordinate the use of OSINET, an experimental computer network for OSI standards, to test for interoperability between systems. OSINET will enable the cooperating organizations to build and verify test systems, conduct company-to-company testing, and to carry out OSI-related research. Also, continued cooperation with other users such as the MAP/TOP Users to arrive at a coordinated government/industry strategy for testing.

ASSOCIATED COSTS?

An indeterminant quantity added to the cost of procurements. It will be a necessary expenditure to assure expensive, complex systems work properly.

F. Raster Compression.

1. WHAT ARE THE STANDARDS? CCITT Recommendation T.6 - 1984

WHAT IS THE CURRENT STATUS OF TESTING?

CCITT Recommendations T.20 and T.21 provide three resolution test charts that may be used for qualitative evaluation of capabilities for (lumped or bundled) transmitters, communications media, and receivers.

WHO'S DEVELOPING THE TESTS?

CCITT

NEED FOR SUPPLEMENTAL TESTS FOR CALS? Yes.

DEFINE THE REQUIREMENTS FOR A TESTING PROGRAM.

Beyond CALS, NBS sees no additional testing program development requirement.

RECOMMEND PROCESS FOR ADMINISTERING TESTING?

Follow CCITT recommendations.

TIMEFRAMES TO ESTABLISH/TO IMPLEMENT?

Not applicable.

SOURCES OF TESTING AND RATIONALE FOR THE RECOMMENDATION(S)?

Manufacturer's Declaration of Conformity

INTERIM SOLUTIONS?

Not Applicable.

ASSOCIATED COSTS?

Not Applicable.

STANDARDS HAVING CALS-SPECIFIC TESTING REQUIREMENTS

A. IGES.

Current status. The concept of application subsets of IGES has been documented to include definition of terms, enumeration of existing efforts, comparison of efforts, identification of requirements for any one subset, draft purchase specifications, test requirements and test cases for assessing conformance.

Application subsets consistent with this strategy have already been defined for technical illustrations, engineering drawings and electronic PC boards. Extensive test cases have been developed for verification of individual entity implementations.

Validation testing requirements for application subsets have been defined for the three subsets already developed. Two software tools have been developed to gain experience with validation testing.

Requirements for acceptance testing rest primarily with the end user to ensure a specific set of IGES translators will work adequately in his environment. Basic guidelines for acceptance testing have been identified but much work is still needed.

Requirements for a Testing Program. A national testing program aimed at IGES verification testing is being set up under the Society of Automotive Engineers. A memorandum of agreement has been signed and test strategy and review procedures are being developed.

Administration of Tests. The verification testing strategy includes the documentation requirements for each test case, the objective of each test, the method for analyzing results and the criteria for measuring success. This testing can be done by a contractor of the national testing program by a vendor of a CAD system or by an end user. In all likelihood, the testing will be done at all three sites.

B. CGM.

Current Status. A CALS application profile will be developed by the end of this fiscal year. In addition, CGM generators and interpreters are being specified for CALS. No software nor specification is developed to date.

Requirements for a Testing Program. Need a CALS test suite to ensure the application profile has been implemented correctly, i.e., all primitives needed by CALS are in the CGM implementation. There would be no need for special hardware.

Administration of Tests. Initial evaluation would recommend

manufacturer's declaration of conformity, third party vendor, or a DoD facility sponsored by OSD as possible candidates (e.g., Wright-Patterson Air Force Base, NSIA consortium).

C. SGML.

Current Status. NBS is developing an SGML validation suite which is to be used to validate SGML parsers. The test suite is expected to be completed by August 1987.

Requirements for a Testing Program. There are at least two kinds of testing to be done: conformance to SGML (whether the parser works correctly) and conformance to military standards which are used to create the document type definitions (document structure). In the latter, conformance is a matter of syntax; if the document has been constructed according to the rules of SGML, it is compliant. Up to a point, NBS can determine a document's conformance or nonconformance by inspection. For CALS, document type definitions need to be developed for the appropriate military standards (e.g., MIL-STD-38784B).

In contrast to document conformance which is described structurally, parser conformance is described functionally. The essential requirement for a parser is that it accept any document as input and inform the user if it cannot determine its underlying structure and content in accordance with the rules of SGML.

Administration of Tests. NBS test suite development will be paid by CALS and collaboration will occur with implementors of SGML software. NBS recommends a manufacturer's declaration of conformity program for SGML validation when the software is complete.

D. ODA/ODIF.

Current Status. Testing development for ODIF awaits ODA implementation availability. Separate CALS conformance testing may be required if OSD desires conformance testing for specific military documents.

E. GOSIP.

Current Status. NBS and DoD will use OSINET to develop gateways between current DoD protocols and OSI protocols. This work may lead to additional testing requirements identified for CALS; however, CALS does not intend on creating a "CALS network." OSINET will also be used to prepare for a demonstration of MAP and TOP products based on OSI protocols. With coordination through NBS, an OSI Lab is currently being established at the Washington Navy Yard. The Corporation for Open Systems is developing a test capability that may be applicable. Some test

capability should be available by the summer of 1988.

F. IRDS.

Current Status. Although not yet identified, NBS anticipates a requirement for a CALS IRDS with specific data elements. If so, conformity testing for a CALS application may be necessary.

G. CCITT, T.6.

Current Status. A program for conformance testing has been developed. NBS will be the interim testing authority for CALS during a four month (Jun-Sep 87) demonstration effort. NBS has issued a purchase order to acquire compression/decompression software for the VAX, operating under VMS, from Delta Information Systems.

Limitations of NBS Testing Facility. The CCITT Recommendation T.6 compression algorithm conformance testing program that is being set up will validate the algorithm for raster scanned data obtained from 8.5" x 11" size paper (American National).

Testing Program. NBS has developed the following procedures for testing:

- A Client submits a written request to NBS.
- NBS sends instructions and the page images to be tested to the Client.
- The Client will send back to NBS a magnetic tape containing the raster scanned compressed images.
- NBS will process the tape decompressing the images and visually compare the results with the original image.

Administration of Tests. As an interim solution, NBS recommends DoD use CCITT recommendations T.20 and T.21 resolution test charts. After the demonstration period, a permanent site must be selected for performing the conformance testing. One of the EDMICS/DESREDS/EDCARS sites is a potential candidate for being an accredited testing laboratory.

TESTING SITE CANDIDATES

Beyond NBS NVLAP accredited testing laboratories as potential candidate sites for conformance testing, other preliminary recommendations include National Laboratories (e.g., Livermore National Laboratory), DoD Service testbeds, trade associations, contractors associated with the Service weapon system demonstrations or impartial third parties.

INDUSTRY-FUNDED COOPERATIVE ARRANGEMENT

Industry could be a major initiator and supporter for acceleration of conformance testing for those standards needed within the CALS environment. Support could be provided in any or all of the following ways:

- a. Volunteer representation on the National and International Standards Committees. (Refer to Appendix C for more information on standards organizations.)
- b. Canvassing "Cooperative" membership and other trade associations for financial and site support. Trade associations such as NCGA, AIA, and SNAME provide a wealth of expertise and facility potential for continuous testing laboratories.
- c. Examine the Service Weapon System Demonstration Projects from an industry perspective, and make recommendations on potential testbeds (e.g., ATF, LHX, JVX, SSN-21, V-22).
- d. Identify potential testing laboratory candidates in the academic community based on past experience.
- e. Develop recommendations and proposed procedures on the most appropriate way to maintain accurate tests and continued testing laboratories for CALS-specific subset applications.
- f. Coordinate industry R&D CALS efforts.
- g. Develop specific CALS applications and compliance tests for standards.
- h. Provide common front to vendor communities (HW, SW, ADP systems).
- i. Accelerate development of CALS technologies, e.g., RAM.

STANDARDS ORGANIZATIONS

Documents related to international database, data dictionary, graphics, and office systems/document interchange standards are available, prepaid, from the following --

American National Standards Institute
International Sales Department
1430 Broadway
New York, NY 10018
Telephone: (212) 642-4900

POSIX (a trademark of IEEE) is the proposed standard for a C language interface to Posix-based portable operating system environments. It is available from:

IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
Telephone: 1-800-272-6657 (Orders Only)
714-821-8380 (Customer Service)

Status Abbreviations:

WD	International Working Draft
DP	Draft Proposed International Standard
DIS	Draft International Standard
IS	International Standard

<u>Status</u>	<u>Document No.</u>
---------------	---------------------

Documents related to computer graphics:

Graphical Kernel System	IS	7942
Computer Graphics Metafile	DIS	8632
GKS-3D	DP	8805
PHIGS	DP	ANSI X3.144

Documents related to data management software and formats:

Information Resource		TC97/SC21
Dictionary System	WD	N473-N477
Database Language NDL	DIS	8907
Database Language SQL	DIS	9075
Data Descriptive File (DDF) for Information Interchange	IS	8211

Documents related to office systems/document interchange:

Office Document Architecture (ODA)
and Interchange Format

Document Structures (Part 2)	DIS	DIS 8613/2
Office Document Interchange Format (Part 5)	DIS	DIS 8613/5
Standard Generalized Markup Language	IS	IS 8879

Documents related to product data interchange:

DPANS (Digital Representation of Product Definition Data)	Awaiting Final Approval by ASME (Y14.26), 7/86
--	---

IGES (NBS)	Version 3.0 4/86
IGES (NBS)	Version 4.0 8/87
PDES (NBS)	Initiation Activities 4/86
PDES (NBS)	Initial Testing Draft 3/87

TERMS AND DEFINITIONS

ACCEPTANCE TESTING. Formal testing conducted to determine whether a software system satisfies its acceptance criteria and to enable the customer to determine whether to accept the system. Formal testing includes the planning and execution of several kinds of tests, (e.g., functional, volume, performance tests) to demonstrate the implemented software satisfies the customer requirements for the software system.

ACCREDITATION. The formalized initial and continuing acceptance of a testing method and or testing laboratory.

ACCREDITED LABORATORY TEST REPORT. A test report which includes a statement by the testing laboratory that it is accredited for the test reported and that the test has been performed in accordance with the conditions prescribed by the accrediting body.

ACCREDITED TEST METHOD. An organized system under which, on a uniform and equitable basis, similar products or services of any number of producers or suppliers may be certified to specified standards.

ASN.1 (Abstract Syntax Notation One) is a standard for data interchange to provide a mechanism for data structures, such as structured databases and files, to be easily moved from one computer system to another.

ASSESSORS. *Selected to conduct an on-site assessment of a particular laboratory on the basis of how well their individual experience matches the type of testing to be assessed. The laboratory has the right to appeal the assignment of an assessor and may request an alternate.

CALS (Computer Aided Logistic Support). An Office of Assistant Secretary of Defense (Acquisition & Logistics) Program concerned with the development and procurement of digital weapon system technical information in all phases of the life cycle.

CERTIFICATE OF CONFORMITY. A document attesting that a product or a service is in conformity with specific standards or technical specifications as determined through use of a specified test method.

CERTIFICATION. The procedure by which a product(s) or service(s) becomes certified.

CERTIFICATION BODY. An impartial body, governmental or non-governmental, possessing the necessary competence and reliability to operate or accredit operation of a

certification system, and in which the interests of all parties concerned with the function of the system are represented.

CERTIFICATION MARK. The Certification Body's validating sign, symbol, or letter that identifies a product(s) or service(s) as being certified.

CERTIFICATION SYSTEM. A system having its own rules of procedure and management, for carrying out conformity certification.

CERTIFIED. Attested by the manufacturer/vendor under the procedures of an accredited testing laboratory as satisfying the requirements of the referenced standard(s).

CGI (Computer Graphics Interface) is a standard designed to specify exchange of device-independent information at the Virtual Device Interface (VDI), which is internal to the graphics system. CGI should be used to support the device-independent transfer requirements of graphic data.

CGM (Computer Graphics Metafile) serves to capture the descriptions of pictures at the level of the CGI; CGM represents a snapshot of the final image that a program has created. CGM is appropriate for CALS in the following situations: viewing the image on a wide variety of devices, enhancing picture qualities, or composing or overlaying several drawings into a single picture for viewing. Since a raster scanning capability will be required in CALS, CGM can also be used to exchange pure raster images.

Client. As used in this Plan, Client refers to any organization or person who employs a testing laboratory for any purpose. Thus, "Client" can refer to a commercial or Certification Body who uses the services of the testing laboratory. A Client is someone who wants products to be tested.

CONFORMITY CERTIFICATION. The action of certifying by means of a certificate of conformity or mark of conformity that a product or service is in conformity with specific standards or technical specifications.

CONFORMITY TESTING. Conformity testing constitutes the testing of a candidate product for the existence of specific characteristics required by a standard. For the purposes of this Strategic Plan, "conformity" and "conformance" will be used interchangeably.

CONFORMITY WITH STANDARDS OR TECHNICAL SPECIFICATIONS. The conformity of a product or a service with all the

requirements of specific standards or technical specifications.

DDF (Data Descriptive File for Information Interchange) is a standard for data interchange to provide a mechanism for data structures, such as structured databases and files, to be easily moved from one computer system to another.

DIF (Document Interchange Format) is a Navy standard for digitized textual interchange between different word processing systems.

EDIF (Electronic Design Interchange Format) is a standards effort in the area of integrated circuit products.

EVALUATORS. *Selected to provide a second opinion, if necessary, and to review the record including the application, assessment report, deficiencies, corrections to deficiencies, and proficiency test results and, based on this record, to recommend whether accreditation should be granted.

FALSIFICATION TESTING. Method using sample cases that test as many of the requirements of the standard as are feasible. The test suite tries to find errors in the implementation. If errors are found, one can correctly deduce the implementation does not conform to the standard; however, the absence of errors does not necessarily imply the converse. The absence of errors implies either that the implementation conforms to the standard or that the test suite was not comprehensive enough to find errors. Falsification testing can only prove nonconformance. (Proofs of Correctness Tests prove conformance to the standard.)

GKS (Graphical Kernel System) consists of nearly 200 user interface routines that give a programmer the ability to create graphical output and accept graphical input from a wide variety of devices. Only 2D primitives are used to describe pictures.

GKS-3D, an upward compatible extension to GKS, will be available within a year.

IGES (Initial Graphics Exchange Specification) is a mature mechanism for the digital exchange of database information among present-day CAD systems. IGES information, including drawings and 3D wireframe product models, is intended for human interpretation at the receiving site. IGES has addressed both mechanical products and printed circuit board products.

IPC (Institute for Packaging of Electronic Components) is a trade association comprised of approximately 900 members. It

is based in Illinois and has a Board of Directors. IPC is an ANSI-approved standards body. It has released standards for quality control over printed circuit boards, and has started doing work in the area of digital data. One area of interest for CALS is the relationship of IGES and IPC standards for manufacturing. From a product file described in IGES format, various IPC standards can be automatically generated to control the manufacturing process for printed circuit boards.

IMPLEMENTATION CONFORMANCE. Defined as implementing at least all the semantics (functions) specified in the standard. Validation tests only concentrate on implementor conformance, and that is what this plan will concentrate on. (Refer to programmer conformance.)

IRDS (Information Resource Dictionary System) contains the specifications for a standard software package that can be used to manage an enterprise's information environment. The IRDS Specifications contain the most commonly used facilities of existing data dictionary systems. The IRDS is an appropriate tool for configuration management and for constructing indexes and directories to data resources (including data in structured databases, graphics databases, paper, microfiche, and other media).

LABORATORY ACCREDITATION. A formal recognition that a testing laboratory is competent to carry out specific tests or specific types of tests.

MANUFACTURER'S DECLARATION OF CONFORMITY. The action by which a manufacturer declares under his sole responsibility, by means of a 'declaration of conformity,' that the product is in conformity with designated standards or other technical specifications, without being under the procedures of a third-party certification system. Note: This term is preferred over self-certification to avoid confusion with the certification process in general.

NDL (is shorthand for Database Language NDL) is a standard for database management systems, and is suitable for high-volume predefined retrievals and updates on databases with stable structures.

ODA/ODIF (Office Document Architecture/Office Document Interchange Format) is an explicit document architecture and interchange format standard which allows exchange of compound documents (i.e., documents composed of various content types, such as character, raster graphics, and geometric (computer) graphics content).

PDES (Product Data Exchange Specification) is focused on exchanging product models with sufficient information content

as to be interpretable directly by advanced CAD/CAM application programs.

PHIGS (Programmer's Hierarchical Interactive Graphics System) is an application programmer's interface to a rich, device-independent graphics environment.

PROGRAMMER CONFORMANCE. Defined as using at most all of the syntax defined in the standard. (Refer to implementation conformance.)

RDA (Remote Data Access) is a standards effort in the area of updating and retrieval of distributed data. Within ISO, the base document is "Standard ECMA-DB, Remote Database Access Service and Protocol, Sixth Draft," of the European Computer Manufacturers Association (ECMA). The report defines: (1) a database model, (2) operations on the database model, and (3) the protocol to support the mapping to the underlying presentation service. This specification is targeted at database systems that support SQL.

REFERENCE MATERIAL. A material or substance one or more properties of which are sufficiently well established to be used for the calibration of an apparatus, the assessment of a measurement method, or for assigning values to materials.

SEMANTICS. The functional description, it defines precisely what must be done, but not how it is to be done (as does the syntax). Generally specified in narrative form using the English language.

SGML (Standard Generalized Mark-Up Language) is a representation language for character text which can be used for CALS publishing applications. The SGML user implicitly defines a document architecture by identifying the components of the document.

SQL (is shorthand for Database Language SQL) is a standard for database management systems, and is suitable for highly flexible retrievals and updates on databases with volatile structures.

SYNTAX. Can consist of verbs in a programming language to access the function or, in the case of graphics standards, the "bindings" to existing programming languages to access the graphics functions in the most natural way for programmers, depending on the language they are using.

TECHNICAL EXPERTS. *Respected peers in their field used as assessors and evaluators, and selected through evaluation of their professional/academic achievements, experience in the field of testing, management awareness, potential for conflict-of-interest, and tact in dealing with people.

TEST METHOD. A defined technical procedure to determine one or more specified characteristics of a material or product. Note 1: Test method includes the test software. Note 2: Part of such a technical procedure may be described by statements in some programming language. The relevant software package and any hardware needed (including reference material) is understood to be part of the test method.

TEST REPORT. A document which presents the test results and other information relevant to the test.

TEST SUITE. The implementation of the test method.

TESTING LABORATORY. A laboratory which measures, examines, tests, calibrates, or otherwise determines the characteristics or performance of materials or products. Note: "Testing Laboratory" in the context of this document may mean (1) a body corporate, (2) one of its subdivisions, (3) the laboratory proper (office and equipment), or (4) the testing service functions for a specific standard.

VALIDATION. Determination of the correctness of the program or software produced to support a specific standard or set of standards.

VERIFICATION. The demonstration of consistency, completeness, and correctness of the product software.

VHSIC Hardware Definition Language (VHDL) is a standards effort in the area of integrated circuit products.

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET <i>(See instructions)</i>		1. PUBLICATION OR REPORT NO. NBSIR 88-3726	2. Performing Organ. Report No.	3. Publication Date MARCH 1988
4. TITLE AND SUBTITLE A Collection of Technical Studies Completed for the Computer-aided Acquisition & Logistic Support (CALS) Program - Fiscal Year 1987 (Vol. 1 of 4)				
5. AUTHOR(S) Edited by Sharon J. Kemmerer				
6. PERFORMING ORGANIZATION <i>(If joint or other than NBS, see instructions)</i> NATIONAL BUREAU OF STANDARDS U.S. DEPARTMENT OF COMMERCE GAITHERSBURG, MD 20899			7. Contract/Grant No.	8. Type of Report & Period Covered NBSIR, 10/86 through 9/87
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS <i>(Street, City, State, ZIP)</i> Office of Assistant Secretary of Defense (A&P)/WSIG Department of Defense, Pentagon Washington, DC 20301-8000				
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.				
11. ABSTRACT <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i> The overall objective of the Department of Defense Computer-aided Acquisition & Logistic Support (CALS) Program is to integrate the design, manufacturing, and logistic functions through the efficient application of computer technology. The National Bureau of Standards has been funded since Spring 1986 to recommend a suite of industry standards for system integration and digital data transfer, and to accelerate their implementation. A major FY87 thrust was the completion of initial documentation of the high-priority standards required in the CALS environment. This volume is one of four providing a collection of the final reports presented to the CALS Policy Office. Major areas contained within this volume include: text, data management, media, raster compression, and conformance testing strategy. The other three volumes contain the graphics and product data reports.				
12. KEY WORDS <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> CALS; conformance; DoD; DOCS IRDS; media; ODA/ODIF; POES raster compression; SGML; SQL; testing				
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 426	
			15. Price \$36.95	