ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

ODA

DOCUMENT SPECIFICATION LANGUAGE

TR/41

July 1987

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône – 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

ODA

DOCUMENT SPECIFICATION LANGUAGE

TR/41

July 1987

TABLE OF CONTENTS

1. GENERAL 1

1.1 Scope 1
1.2 Field of Application 1

2. DEFINITIONS AND REFERENCES 1

2.1 Definitions 1

2.1.1 Document Script 1

2.2 Acronyms 1
2.3 References 1

3. OVERVIEW 2

3.1 Introduction 2
3.2 General Principles 2

4. DESCRIPTION OF THE METALANGUAGE 3

4.1 Syntax 3
4.2 Keywords 3
4.3 Production Names 3
4.4 Literal Names 3
4.5 Optional Element Sequences 3
4.6 Element Sequence Choices 3
4.7 Comments 3

5. SPECIFICATION LANGUAGE SYNTAX 3

5.1 Document 4
5.2 Document Specification 4
5.3 Logical Definitions 5
5.4 Layout Definitions 9
5.5 Logical Structure 10
5.6 Layout Structure 12
5.7 Content Portions 13
5.8 Character Presentation Attributes 14
5.9 Photographic Presentation Attributes 16
5.10 Expressions 17
5.11 Constructions 17
5.12 Defaults 18
5.13 Content Architecture 18
5.14 Names 18
5.15 Cardinal 18
5.15 Character Strings 18

6. CONVENTIONS FOR CONTROL FUNCTIONS 18

7. COLLECTED SYNTAX 21

7.1 document 21
7.2 document-specification 22
7.3 logical-definitions 23
7.4 logical-definition 23
7.5 layout-definitions 24
7.6 layout definition 24
7.7 logical-structure 25

7.8 logical-object 25
7.9 layout structure 26
7.10 layout-object 26
7.11 content-portions 27
7.12 content-portion 27
7.13 character-presentation-attributes 28
7.14 photographic-presentation-attributes 29
7.15 conte nt-architecture 29
7.16 measure 29
7.17 expression 29
7.18 string-expression 29
7.19 atomic-string-expression 29
7.20 cardinal-expression 30
7.21 object-id-expression 30
7.22 object-selection-function 30
7.23 construction 30
7.24 construction-expression 30
7.25 construction-term 30
7.26 clo-defaults 31
7.27 blo-defaults 31
7.28 page-set-defaults 31
7.29 page-defaults 31
7.30 frame-defaults 32
7.31 block-defaults 32

APPENDIX A - GUIDELINES FOR PRESENTATION 33

APPENDIX B - EXAMPLES 35

APPENDIX C - LIST OF KEYWORDS 49

1. GENERAL

1.1 Scope

This Technical Report defines a language for the human readable representation of information structured according to Standard ECMA-101, Office Document Architecture.

This Technical Report defines the meta syntax used for formally specify the language; defines the language syntax and semantics, and provides a collected syntax for reference use.

The two appendices to the report provide guidelines for use in terms of the language by two examples.

1.2 Field of Application

The language defined by this Technical Report has two primary applications.

Firstly, it provides an unambiguous representation of ODA documents and document classes which is legible and intelligible to people.

Secondly, it provides a means for designing ODA document classes which may then be directly used by automatic document preparation equipment, such as word processors and document processors.

The specification language is not intended for interchange, and as such it is not an alternative to ODIF, which is the standard representation of ODA documents and document classes used for interchange.

2. DEFINITIONS AND REFERENCES

2.1 Definitions

For the purpose of this Technical Report, the following definitions apply. These definitions supplement those in Section 2.1 of the ODA Standard, ECMA-101.

2.1.1 Document Script

An instance of the use of the Specification Language in describing an ODA document.

2.2 Acronyms

BMU	Basic Measurement Unit
BNF	Backus Naurer Form
ISO	International Organization for Standardization
ODA	Office Document Architecture
ODIF	Office Document Interchange Format

2.3 References

ECMA-17	Graphic Representation of the Control Characters of the ECMA 7-Bit Coded Character Set for Information Interchange
ECMA-101	Office Document Architecture
ISO 2014	Writing of calendar dates in all-numeric form
ISO 3307	Representations of time of the day
ISO 4031	Representation of local time differentials
ISO 6937	Coded character sets for text communication

3. OVERVIEW

3.1 Introduction

Standard ECMA-101 states that it is to be applied to the interchange of office documents such as memoranda, letters, forms, etc. Information coded according to ECMA-101 is intended for communication between machines such as word processors or document processors.

The language defined in this Technical Report is intended to be used directly by people as a method of representing similar types of documents to those identified by ECMA-101.

The primary application of this language is the communication of information between people and thus it has a human readable representation. However, whilst this language is not proposed as a method of communication of office documents between machines, it may be used to communicate between people and machines. That is, it has a format which is amenable to both human comprehension and machine processing.

This duality of capability means that it may be used both for communication between people, for applications such as education, and for machine-oriented applications, such as the design of ODA document classes.

Throughout this specification, an instance of the use of the Specification Language describing an ODA document is termed a script or document script.

3.2 General Principles

The general principles to which this specification language adheres are as follows.

- The language does not adhere strictly to the ODA form of attribute representations, but there is a simple correspondence between the two representations.
- The language is English-oriented, avoiding the use of special symbols, and thus aiding readability.
- Meaningful names are used instead of the formal component identifiers used within the ODA Standard.
- The grammar is simple, to aid the user in composition and to simplify machine compilation.
- Alternative measurement units may be defined, since BMUs are not likely to be directly meaningful to the human reader. These units are, however, defined in relation to BMUs.
- The language itself comprises only characters within the minimum subrepertoire of ISO 6937/2. This ensures that the language be preparable on minimum capability equipment.
- A number of ODA attributes have no direct representation in the language, since they can be derived from other properties.

The description of the structure and content of an ODA document given in this Technical Report is not intended to be definitive. Where the text herein is not sufficiently clear, or where more detail is required, the reader should refer to the ODA Standard, ECMA-101.

4. DESCRIPTION OF THE METALANGUAGE

4.1 Syntax

This section formally defines the meta-language used throughout this report to describe the ODA Specification Language.

The meta-language has the following BNF representation.

4.2 Keywords

A keyword is a reserved sequence of characters, the meaning of which is defined by the specification language. Keywords are composed of capital letters and hyphen. A complete list of keywords is given in Appendix C.

4.3 Production Names

A production-name is the name of a syntactic rule. Production names are composed of small letters and hyphen.

```
<document>                 ::=   <element-seq>
<element-seq>              ::=   <element> |
                                 <element> <element-seq>
<element>                  ::=   <keyword>
                                 <production-name>
                                 <literal-name> |
                                 <optional-element> |
                                 <element-choice> |
                                 <element-choice> "..."
<optional-element-seq>     ::=   "[" <element-seq> "]"
                                 "{" <element-seq> "}"
                                  {  <element-seq>   }
<element-choice-seq>       ::=   {       ...         }
                                  {  <element-seq>   }
```

4.4 Literal Names

A literal-name is the name of an element the syntax of which is not formally defined but the construction of which is described in the accompanying text. Literal names are composed of small letters and hyphen, and are italicized.

4.5 Optional Element Sequences

An optional-element-seq means that the element sequence is optional and therefore may or may not be present in a particular document script.

4.6 Element Sequence Choices

An element-seq-choice means that one of the element sequences is to be chosen. An element sequence choice may be followed by "..." which means that it may be repeated.

4.7 Comments

Comments may be included at any point in a document script. A comment starts with /* and terminates with */. Comments may be nested to any depth.

5. SPECIFICATION LANGUAGE SYNTAX

This section defines the syntax and semantics of the ODA Specification Language. The syntax is described using the notation of 4.

The description of the syntax is divided into subsections. Each subsection covers a major topic in the language. Each of these topics closely follows the subdivision of the ODA Standard, namely:

- language features describing the document as a whole,
- logical structure,
- layout structure,
- content.

5.1 Document

This section provides information to identify the document script and which is used throughout the remainder of the script.

The ODA DOCUMENT statement identifies the document script as being an ODA document. There is no ODA attribute corresponding to this statement

The LANGUAGE VERSION statement identifies the version of the Specification Language in which the document script is written.

The STANDARD VERSION statement identifies the version of ODA to which this script conforms. This statement is represented by the attribute "ODA Version Date" within the document profile.

The FONT NAMES statement specifies the list of character fonts used in the script. This statement is represented by character fonts information within the document profile.

The MEASUREMENT UNITS statement specifies names units used in the script together with their relationship to Basic Measurement Units. The statement enables distances and size values to be specified in more conventional units such as inches or millimetres.

5.2 Document Specification

This section comprises the bulk of the document script. It commences with a sequence of statements which provide information which is used within the document profile, and continues with the document body itself.

The DOCUMENT statement specifies an identifier to be associated with the document as assigned by the author of the document script. It does not correspond to any ODA attribute.

The TITLE statement specifies a short description assigned by the author in order to identify the document. It corresponds to the document profile attribute "Title".

The VERSION statement specifies a number associated with the document to distinguish it from other documents bearing the same reference name. It corresponds to the document profile attribute "Version Number".

The KEYWORDS statement specifies logical associations to the content of the document. It corresponds to the document profile attribute "Keywords".

The REFERENCE statement specifies a label (reference number) assigned by the author for the identification of the document within a document library. It corresponds to the document profile attribute "Reference".

The DATED statement specifies a date associated with the document by its author or originating organization. The date is expressed in accordance with ISO 2014 and ISO 3307. It corresponds to the document profile attribute "Document Date".

The EXPIRES ON statement specifies a date after which the document is considered invalid by the originator. The date is expressed in the same manner as for DATED above. It corresponds to the document profile attribute "Expiry Date".

The CLASSIFICATION statement specifies the type of document (memorandum, letter, report, etc) as assigned by the originator. It corresponds to the document profile attribute "Document Class".

The ORIGINATORS INFORMATION phrase specifies properties of the document pertaining to the origin and nature of the document.

The ORGANIZATIONS statement specifies the originating organization associated with the document. It corresponds to the document profile attribute "Organizations".

The PREPARED BY statement specifies the name of the person responsible for the preparation and/or encoding of the document. It corresponds to the document profile attribute "Preparer".

The OWNED BY statement specifies the current administrators of the document. It corresponds to the document profile attribute "Owners".

The AUTHORS statement specifies the authors of the document. It corresponds to the document profile attribute "Authors".

The COPYRIGHT statement specifies the name of the legal party in whom the Copyright of the document is vested. It corresponds to the document profile attribute "Copyright".

The STATUS statement specifies the status of the document (working draft, approved, superseded, etc). It corresponds to the document profile attribute "Status".

The DISTRIBUTION statement specifies the recipients of the document as intended by the originator. It corresponds to the document profile attribute "Circulation List".

The OTHER INFORMATION statement specifies additional user-specified information relevant to the document (contract number, project number, budget code, etc). It corresponds to the document profile attribute "User-specific Codes".

The subsequent five phrases (logical definitions, layout definitions, logical structure, layout structure and content portions) constitute the document body itself, and are described in the following sections.

The END DOCUMENT statement indicates the end of the document script.

5.3 Logical Definitions

The logical definitions phrase specifies the generic logical structure of the document script.

The logical definitions phrase consists of a sequence of logical definitions. Each logical definition represents an ODA logical class description.

Logical definitions may be present in the script in any order. However, the ODA rules concerning class definitions apply, viz. each class definition referenced in the COMPOSED OF statement must have a corresponding logical definition.

It is expected that logical definitions will be written in the document script in an order similar to the sequential logical order defined by the ODA Standard (see ECMA-101, 4.1.1).

The logical definition phrase commences at the DEFINES statement and terminates at the corresponding END statement.

The DEFINES statement specifies the name of the logical definition. This name can subsequently be used in other statements, such as the COMPOSED OF statement, in order to refer to it. The DEFINES statement also specifies whether the logical object is of the type DOCUMENT, COMPOSITE or BASIC, and if it is protected.

The elements of the DEFINES statement correspond to ODA attributes as follows:

- definition name corresponds to "Definition Identifier"
- PROTECTED corresponds to "Protected"
- The keywords DOCUMENT, COMPOSITE and BASIC correspond to similarly named values of the attribute "Object Type".

The END statement specifies the end of the information describing the logical definition with the specified name.

The ANNOTATED WITH statement specifies text that can be used as a description of the logical definition. It is represented by the attribute "User Readable Comment".

The LAYOUT statement specifies whether the content of the logical definition is to commence in a new basic layout object, by the STARTING keyword, or whether it is to be concatenated with the preceding content, by the CONTINUING keyword.

The CATEGORISED BY statement specifies the category name of the layout object with which this logical object is to be associated. This statement is represented by the attribute "Required Category".

The WITHIN NEW statement specifies that the content of the logical object is to commence in a new layout object as identified by the named definition or category. This statement is represented by the attribute "New Layout Object".

The EXCLUSIVELY WITHIN statement specifies that the content of the logical object is to be contained within a single instance of the layout object as identified by the named definition, and that no other content is to be contained within that layout object. This statement is represented by the attribute "Layout Reference".

The UNDIVIDED WITHIN statement specifies that the content of the logical object is to be contained in a single instance of a layout object as identified by the named definition or category. Unlike the EXCLUSIVELY WITHIN statement, other content may be contained within the same layout object. This statement is represented by the attribute "Indivisible".

The BOUNDED BY statement specifies the distances between the layout object containing the content of the logical object and its containing layout object. The meaning of this statement is illustrated in Figure 1.

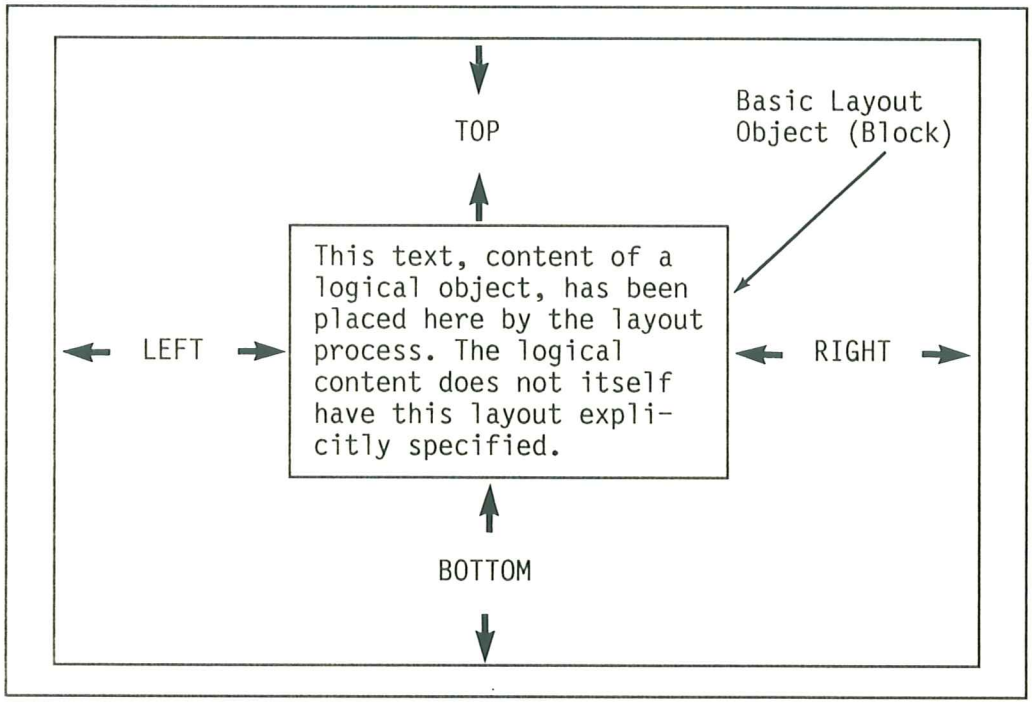


Figure 1 - The effect of the BOUNDED BY statement

The SEPARATED BY statement specifies the minimum distances between the basic layout object containing the logical object and the nearest adjacent basic layout objects. The distances are specified in terms of the LEADING and TRAILING edges of the containing basic layout object. LEADING and TRAILING are themselves defined in terms of the BLOCK PROGRESSION statement. The meaning of the SEPARATED BY statement is illustrated in Figure 2 (next page). This statement is represented by the attribute "Separation".

The DEFAULTS statement specifies the values that are to be considered to exist for statements which may optionally be present in a logical definition phrase. It specifies the statement values which are to apply to composite and basic logical objects by the FOR COMPOSITE OBJECT and BASIC OBJECT constructs. These specify the values to apply via the clo-defaults and blo-defaults phrases respectively. The language constructs for these are described in 5.12.

The LOCAL DATA statement specifies named expressions used to determine values associated with the logical object. The data values associated with a logical object each have a name, and an expression used to determine the value. In general, an expression can be a numeric or string constant or a local data reference. A full description of expressions is given in 5.10. This statement is represented by the attribute "Bindings".

The CONTENT phrase specifies properties describing the content of the logical object, how it is to be presented, and, finally, the content itself.

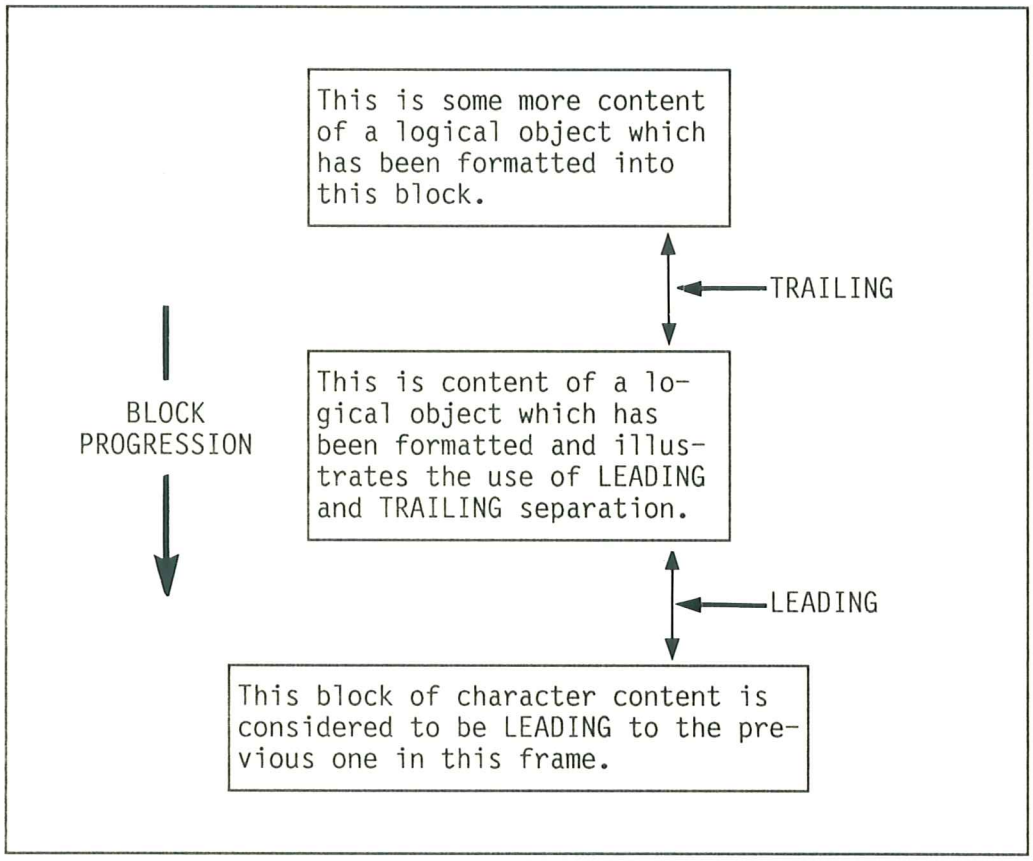


Figure 2 - The effect of the SEPARATED BY statement

The properties describing the content are specified by content architecture and content type. Content type is specified by one of the two keywords CHARACTER and PHOTOGRAPHIC. content-architecture specifies the particular type of content encoding to be used. The keywords for content-architecture are specified in 5.13. This statement is represented by the attributes "Content Type" and "Content Architecture".

The presentation characteristics of the content are specified by the PRESENTATION phrase. The presentation characteristics specific to the content type follow the keyword PRESENTATION. Presentation characteristics specify, for example, that characters are to be justified, centred, etc. A full description of character and photographic presentation attributes is given in 5.8 and 5.9, respectively.

The CONTAINING statement specifies the content to be associated with the logical definition. The content may be specified in one of three ways:

- as a character string (the format of character strings is described in 5.16),
- as a string expression,
- as a sequence of named content portions.

The first method is the usual method of specifying character content. The second is used when the content is composed of content which, for example, has computed values. The third form is used when the content is, for example,

used by several logical definitions, and hence, by using a single instance of the actual content, this method reduces the size of the document script. This third method is also the only method of specifying photographic content.

5.4 Layout Definitions

The layout definitions phrase specifies the generic layout structure of the document script. It consists of a sequence of layout definitions, each of which represents an ODA layout class description.

Layout definitions may be present in the script in any order. However, the ODA rules concerning class definitions apply, viz. each class definition reference within the COMPOSED OF statement must have a corresponding layout definition.

It is expected that layout definitions will be written in the document script in an order similar to the sequential layout order defined by the ODA Standard (see ECMA-101, 4.1.1).

The layout definition phrase commences with the DEFINES statement and terminates with the corresponding END statement.

The DEFINES statement specifies the name of the layout definition. This name can subsequently be used in other statements such as the COMPOSED OF statement. The DEFINES statement specifies whether the layout object is of the type DOCUMENT, PAGE-SET, PAGE, FRAME or BLOCK. The keywords of the DEFINES statement correspond to ODA attributes as follows:

- definition-name corresponds to "Definition Identifier",
- the keywords DOCUMENT, PAGE-SET, PAGE, FRAME and BLOCK correspond to the similarly named values of the attribute "Object Type".

The END statement specifies the end of the information describing the layout definition with the specified name.

The ANNOTATED WITH statement specifies the text that can be used as a description of the layout definition. It corresponds to the attribute "User Readable Comments".

The POSITIONED AT statement consists of a pair of coordinates that specify the position of the object relative to the object at the next higher level in the hierarchy (i.e. either the containing page or frame). Each coordinate measure is composed of a cardinal number followed by the unit name which can be in BMUs or any other unit defined at the beginning of the document script (see 5.1). This statement corresponds to the attribute "Position".

The WIDTH AND HEIGHT statement consists of a pair of dimensions that specify the size of the object. Dimensions are specified in the same manner as the coordinates of the preceding statement. This statement corresponds to the attribute "Dimensions".

The COMPOSED OF statement specifies the rule for constructing instances of the layout definition. It consists of an expression. This statement is represented by the attribute "Generator for Subordinate Objects".

The BLOCK PROGRESSION statement specifies the direction in which the blocks are to be laid out within the layout object. This statement corresponds to the attribute "Object Path".

The ACCEPTING CATEGORIES statement specifies the layout categories permitted for logical objects which are to be laid out within a layout object derived from this definition. Category names are declared in corresponding logical definitions by the CATEGORISED BY statement. The ACCEPTING

CATEGORIES statement corresponds to the relational attribute "Permitted Categories".

The BACKGROUND OPAQUE statement specifies that the frame or block is to be white opaque. Otherwise the frame or block is colourless transparent. This statement corresponds to the property attribute "Background Texture".

The DEFAULTS phrase introduces the default value list for the subordinate layout objects. It corresponds to the attribute "Default Value Lists".

For PAGE-SET, the only specifiable defaults are for referencing an object definition, using the FROM statement described under PAGE-SET above, and for specifying dimensions, as described under the WIDTH AND HEIGHT statement above.

For PAGE, the specifiable defaults are for referencing an object definition, using the FROM statement described under PAGE-SET above, and for specifying dimensions, as described under the WIDTH AND HEIGHT statement above.

For FRAME, the specifiable defaults are for the statements FROM, POSITIONED AT, WIDTH AND HEIGHT, BLOCK PROGRESSION, ACCEPTING CATEGORIES, and BACKGROUND OPAQUE. These statements are described elsewhere in this Technical Report.

For BLOCK, the specifiable defaults are for the statements FROM, POSITIONED AT, WIDTH AND HEIGHT, BACKGROUND OPAQUE, and PRESENTATION. Once again, these statements are described elsewhere.

The LOCAL DATA statement specifies named expressions used to determine values associated with the layout object. The data values associated with a layout object each have a name, and an expression used to determine the value. In general, an expression can be a numeric or string constant, or a local data reference. A full definition of expressions is given in 5.10. The LOCAL DATA statement is represented by the attribute "Bindings".

The CONTENT phrase specifies properties describing the logical content, how it is to be presented and, finally, the content itself. For more information on this statement, see 5.3. The CONTENT statement corresponds to the presentation attributes "Content Type" and "Content Architecture".

The presentation characteristics of the content are specified by the PRESENTATION statement. For more information on this statement, see 5.8 and 5.9 (Character Presentation Attributes and Photographic Presentation Attributes, respectively).

The CONTAINING statement specifies the content associated with the layout definition. For more information on this statement, see 5.3 and 5.16.

5.5 Logical Structure

The logical structure phrase specifies the specific logical structure of the document script.

The logical structure phrase consists of a single logical object which corresponds to the logical document object. Subordinate logical objects are represented within the SUBORDINATES statement of this phrase. Each logical object represents an ODA logical object description.

The logical definition phrase commences at the IS statement and terminates at the corresponding END statement.

The IS statement specifies the type of the logical object, which may be LOGICAL DOCUMENT, COMPOSITE OBJECT or BASIC OBJECT, and whether or

not the object is protected. This information is represented by the ODA attributes "Object Type" and "Protected", respectively.

The END statement specifies the end of the information describing this logical object.

The ANNOTATED WITH statement specifies text that can be used as a description of the logical object. It is represented by the attribute "User Readable Comment".

The LAYOUT statement specifies whether the content of the logical object commences in a new basic layout object, by the STARTING keyword, or whether it is concatenated with the preceding content, by the CONTINUING keyword.

The CATEGORISED BY statement specifies the category name of the layout object with which this logical object is associated. This statement is represented by the attribute "Required Category".

The WITHIN NEW statement specifies that the content of the logical object commences in a new layout object as identified by the named definition or category. This statement is represented by the attribute "New Layout Object".

The EXCLUSIVELY WITHIN statement specifies that the content of the logical object is contained within a single instance of the layout object as identified by the named definition, and that no other content is contained within that layout object. This statement is represented by the attribute "Layout Reference".

The UNDIVIDED WITHIN statement specifies that the content of the logical objects is contained in a single instance of the layout object as identified by the named definition or category. Unlike the EXCLUSIVELY WITHIN statement, other content may be contained within the same layout object. This statement is represented by the attribute "Indivisible".

The BOUNDED BY statement specifies the distances between the layout object containing the content of the logical object and its containing layout object. The meaning of this statement is illustrated in Figure 1. This statement is represented by the attribute "Offset".

The SEPARATED BY statement specifies the distances between the layout object containing the logical object and the nearest adjacent basic layout objects. The distances are specified in terms of the LEADING and TRAILING edges of the containing basic layout object. LEADING and TRAILING are themselves defined in terms of the BLOCK PROGRESSION statement. The meaning of the SEPARATED BY statement is illustrated in Figure 2. This statement is represented by the attribute "Separation".

The LOCAL DATA statement specifies named expressions used to determine values associated with the logical object. The data values associated with the logical object have each a name, and an expression used to determine a value. In general, an expression can be a numeric or string constant or a local data reference. A full description of expressions is given in 5.8. This statement is represented by the attribute "Bindings".

The SUBORDINATES statement specifies the logical objects directly subordinate to this logical object as a sequence of logical object phrases. The order of logical objects within this statement corresponds to their order within the sequential logical order of the document. The SUBORDINATES statement corresponds to the attribute "References to Subordinate Objects".

The CONTENT phrase specifies properties describing the content of the logical object, how it is to be presented, and, finally, the content itself.

The properties describing the content are specified by content architecture and content type. Content type is specified by one of the two keywords CHARACTER and PHOTOGRAPHIC. content-architecture specifies the particular type of content encoding to be used. The keywords for content-architecture are specified in 5.11. This statement is represented by the attributes "Content Type" and "Content Architecture".

The presentation characteristics of the content are specified by the PRESENTATION phrase. The presentation characteristics specific to the content type follow the keyword PRESENTATION. Presentation characteristics specify, for example, that characters are justified, centred, etc. A full description of character and photographic presentation attributes is given in 5.6 and 5.7, respectively.

The CONTAINING statement specifies the content to be associated with the logical object. The content may be specified in one of two ways:

- as a character string (the format of character strings is described in 5.14),

- as a sequence of named content portions.

The first method is the usual method of specifying character content. The second form is used when the content is, for example, used by several logical objects, and hence, by using a single instance of the actual content, this method reduces the size of the document script. This second method is also the only method of specifying photographic content.

5.6 Layout Structure

The layout structure phrase specifies the specific layout structure of the document script.

The layout structure phrase consists of a single layout object which corresponds to the layout document object. Subordinate layout objects are represented within the SUBORDINATES statement of this phrase. Each layout object represents an ODA layout object description.

The layout definition phrase commences with the IS statement and terminates with the corresponding END statement.

The IS statement specifies the type of the layout object, which may be LAYOUT DOCUMENT, PAGE-SET, PAGE, FRAME or BLOCK. This information is represented by the ODA attribute "Object Type".

The END statement specifies the end of the information describing this layout object.

The ANNOTATED WITH statement specifies the text that can be used as a description of the layout object. It corresponds to the attribute "User Readable Comments".

The POSITIONED AT statement consists of a pair of coordinates that specify the position of the object relative to the object at the next higher level in the hierarchy (i.e. either the containing page or frame). Each coordinate measure is composed of a cardinal number followed by the unit name which can be in BMUs or any other unit defined at the beginning of the document script (see 5.3). This statement corresponds to the attribute "Position".

The WIDTH AND HEIGHT statement consists of a pair of dimensions that specify the size of the object. Dimensions are specified in the same manner as the coordinates of the preceding statement. This statement corresponds to the attribute "Dimensions".

The BLOCK PROGRESSION statement specifies the direction in which the blocks are laid out within this layout object. This statement corresponds to the attribute "Object Path".

The BACKGROUND OPAQUE statement specifies that the frame or block is white opaque. In the absence of this statement, the frame or block is colourless transparent. This statement corresponds to the property attribute "Background Texture".

The LOCAL DATA statement specifies named expressions used to determine values associated with the logical object. The data values associated with the logical object have each a name, and an expression used to determine a value. In general, an expression can be a numeric or string constant or a local data reference. A full description of expressions is given in 5.8. This statement is represented by the attribute "Bindings".

The CONTENT phrase specifies properties describing the logical content, how it is presented and, finally, the content itself. For more information on this statement, see 5.3. The CONTENT statement corresponds to the presentation attributes "Content Type" and "Content Architecture".

The presentation characteristics of the content are specified by the PRESENTATION statement. For more information on this statement, see 5.6 and 5.7 (Character Presentation Attributes and Photographic Presentation Attributes, respectively).

The statement specifies the content associated with the layout definition. For more information on this statement, see 5.3 and 5.14.

5.7 Content Portions

The content portions phrase specifies those content portions with which the author of the document script has associated a name. (Some character content portions may appear directly within the specifications of objects and definitions.) It consists of a sequence of content portion specifications, each of which represents an ODA content portion.

Within the content portions phrase, the content portions may be present in any order. However, it is expected that the order within a particular document script will be that of their appearance in the remainder of that script.

The content portion phrase commences at the IS statement and terminates at the corresponding END statement.

The IS statement specifies the name of the content portion. This name can subsequently be used elsewhere in the document script, such as in the CONTENT phrase of a definition, in order to refer to it. The IS statement also specifies whether the content is character or photographic, and, in the latter case, if it is compressed.

The elements of the IS statement correspond to ODA attributes as follows:

- content-portion-name corresponds to "Content Portion Identifier",
- the keywords CHARACTER and PHOTOGRAPHIC correspond to similarly named values of the attribute "Content Type",
- UNCOMPRESSED corresponds to the coding attribute "Compression" within a photographic content portion.

The WITH phrase specifies information about the coding of a photographic content portion.

The PELS PER LINE statement specifies the number of pels in each line of the photographic image. It corresponds to the coding attribute "Number of Pels per Line".

The PELS DISCARDED statement specifies the number of pels at the beginning of each line which are to be discarded for the purpose of imaging. It corresponds to the coding attribute "Number of Pels Discarded".

The LINES statement specifies the number of lines in the photographic image. It corresponds to the coding attribute "Number of Lines".

For both character and photographic content portions, the content may be stored externally to the document script. In this case, the FROM statement specifies the name of the file in which the content is stored. Such a file name is necessarily system dependent, and thus its construction is not determined within this Technical Report. This is the only means of specifying photographic content.

For a character content portion, the content may alternatively be specified explicitly in the document script as a character string.

5.8 Character Presentation Attributes

The character presentation attributes phrase controls the format and appearance of the character content.

The alignment statement specifies whether the character content is to be JUSTIFIED, CENTRED, LEFT-ALIGNED or RIGHT-ALIGNED. The statement determines the appearance of all the content with which it is associated and cannot be modified by codes present in the content itself. It can be present in both logical and layout definitions. This statement is represented by the character content attribute "Alignment".

The FIRST LINE statement specifies the appearance of the first line of the character content with respect to subsequent lines. It can only be present in logical definitions. The appearance can be specified as INDENTED or OVERHANGING. The BY keyword specifies by how much the first line is to be indented or to overhang the subsequent lines. Indentation and overhang are illustrated in Figures 3 and 4. This statement is represented by the character content attribute "First Line Indentation.

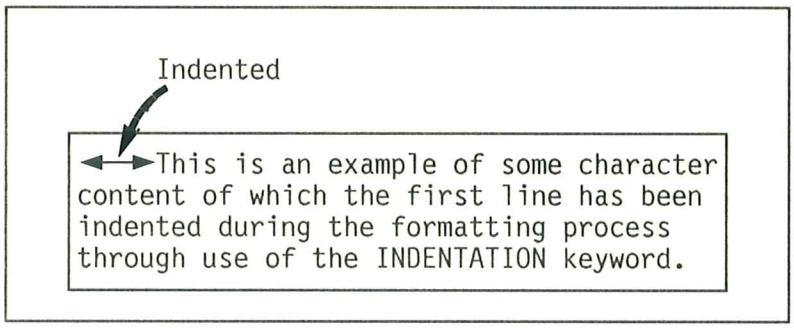


Figure 3 - The effect of the INDENTED keyword

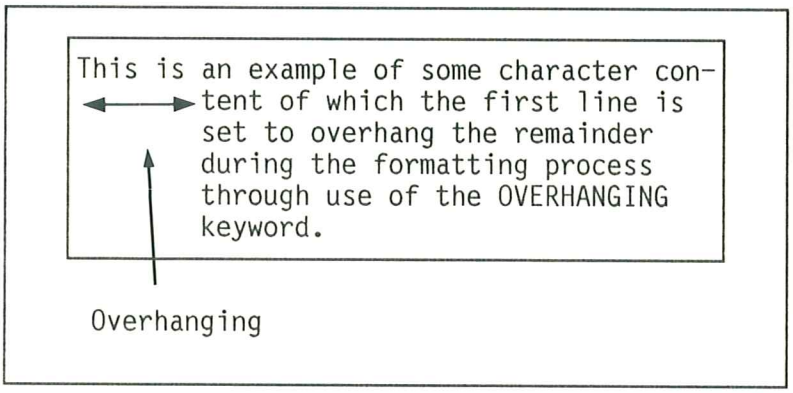


Figure 4 - The effect of the OVERHANGING keyword

The WIDOW statement specifies the minimum number of lines of the associated content which is to occur in the first basic layout object if the content occurs in a sequence of more than one basic layout object. It can only be present in logical definitions.

The ORPHAN statement is similar to the WIDOW statement, but in this case the restriction applies to the last basic layout object rather than the first.

The WIDOW and ORPHAN statements are represented by the character content attributes "Widow Size" and "Orphan Size", respectively.

The TAB STOPS statement specifies points in a line to which successive HT control functions will position the content. In the case that a line of content has more HT control functions than are specified by the TAB STOPS statement, the additional HT control functions are ignored. This statement can specify that the content following an HT is to be LEFT-ALIGNED, RIGHT-ALIGNED, CENTRED or ALIGNED AROUND a specified character string. It is represented by the character content attribute "Line Layout Table".

The TAB STOPS statement may be specified for logical and layout definitions and objects. However, in a script containing both logical and layout structures, it may only be specified for the logical definitions and objects.

The PREFERRED FONTS statement specifies up to ten fonts which can be used within a basic logical or layout object. Which font is to be used at any particular point is determined by the SGR control function within the content. This statement is represented by the character content attribute "Character Fonts", and can be present in both logical and layout definitions and objects.

The CHARACTER ORIENTATION statement specifies the appearance of characters in terms of a rotation with respect to the direction set by the CHARACTER PATH statement. This statement can specify one of four angular rotations. It is represented by the character content attribute "Character Orientation".

The CHARACTER PATH statement specifies the direction of placement, in a basic layout object, of successive characters of the content. This direction is specified as one of four angles with respect to the horizontal edge of the basic layout object. This statement is represented by the character content attribute "Character Path".

The LINE PROGRESSION statement specifies the direction of placement, in a basic layout object, of successive lines of content. This direction is specified by one of two angles with respect to the direction determined by the CHARACTER PATH statement. This statement is represented by the character content attribute "Line Progression".

The INITIALLY USING phrase specifies properties concerning the appearance of character content that are to apply at the beginning of its presentation, and which may be changed by control functions within the content. The control functions affecting the properties of this phrase are specified in their corresponding attribute definition.

The CHARACTER SPACING statement specifies the unit character spacing. The statement only applies to constant spaced character content and not to proportionally spaced content. This statement is represented by the character content attribute "Character Spacing".

The LINE SPACING statement specifies the subrepertoire of the ISO 6937 character set to be used in the content. This statement is represented by the character content attribute "Graphic Subrepertoire".

5.9 Photographic Presentation Attributes

The INITIAL OFFSET statement specifies the reference position of the first photographic element (pel) within the basic layout object. A pair of measures specify the X and Y coordinates of the first pel relative to the origin of the basic layout object (see Figure 5). Each measure is composed of a cardinal number followed by the name of a unit, which can be BMUs or any unit defined at the beginning of the document script (see 5.1). The INITIAL OFFSET statement corresponds to the rendition attribute "Initial Offset".

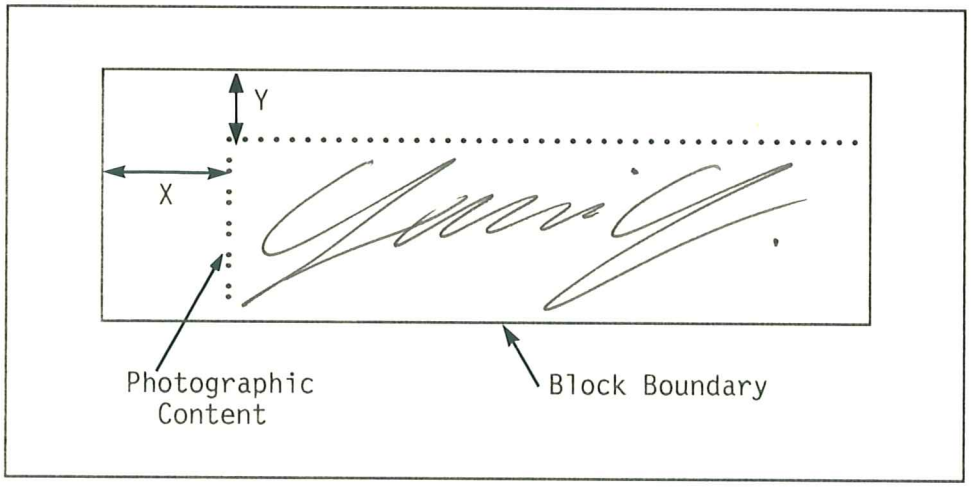


Figure 5 - The effect of the INITIAL OFFSET statement

The PEL PATH statement specifies the direction of progression of successive pels along a line, in terms of the angle between the X axis of the basic layout object and the desired direction. This statement corresponds to the rendition attribute "Pel Path".

The LINE PROGRESSION statement specifies the direction of progression (90 = upward; 270 = downward) of successive lines relative to "Pel Path". This statement corresponds to the rendition attribute "Line Progression".

The UNIFORM RESOLUTION statement specifies the number of pels in the unit length (25,4 mm). A single number specifies both horizontal and vertical resolution. Differing horizontal and vertical resolutions can be specified using the alternative RESOLUTION statement in place of UNIFORM RESOLUTION. This specifies a number of pels in the unit length independently for the horizontal and vertical directions. Both these statements correspond to the rendition attribute "Resolution.

5.10 Expressions

Expression phrases occur in a number of places in the specification language. These expression phrases correspond to expressions within attribute values and may be of one of three types, viz. string expression, cardinal expression or object identifier expression.

The syntax for expressions within the specification language closely mirrors the descriptive text within the ODA Standard. Only those features of expressions peculiar to the specification language are described here, and the reader should refer to ECMA-101, 6.4, for further detail.

Within the options for an atomic string expression and a cardinal expression, the FROM statement specifies that the value should be obtained from the binding with the specified data name on the object with the identifier specified by the object identifier expression. See ECMA-101, 6.4.2, for further detail.

Within the cardinal expression phrase, the INCREMENT and INC statements are synonymous, as are the DECREMENT and DEC statements. They correspond to the INCREMENT and DECREMENT functions described in ECMA-101, 6.4.4.

Within the object identifier expression phrase, the pairs of statements SUPERIOR and SUP, PRECEDING and PREC, and SUCCEEDING and SUCC are all synonymous. They correspond to the SUPERIOR, PRECEDING-ID and SUCCEEDING-ID functions described in ECMA-101, 6.4.1.

5.11 Constructions

The construction phrase is used within the COMPOSED OF statement of a definition to specify the composition of an object derived from that definition in terms of other definitions. It may take one of two forms.

The first, and simpler, form is a list of definition names. This corresponds to a simple construction, as defined in ECMA-101, 6.5.5.2.

The second, more general, form is a construction expression. This is either a SEQUENCE statement, an AGGREGATE statement, or a CHOICE statement, which is composed of the appropriate keyword followed by a sequence of construction terms. These three forms correspond to a sequence construction, an aggregate construction, and a choice construction, respectively. The pairs of keywords SEQUENCE and SEQ, AGGREGATE and AGG, and CHOICE and CHO are synonymous.

A construction term is either the name of a definition or a parenthesised construction expression, optionally preceded by the keywords OPTIONAL and/or REPEAT. These forms correspond to a required construction factor, an optional construction factor, a repetitive construction factor, and an optional repetitive construction factor, respectively. The pairs of keywords OPTIONAL and OPT, and REPEAT and REP are synonymous.

The semantics of construction expressions, construction terms and construction factors are defined in ECMA-101, 6.5.5.2, and are not repeated here.

5.12 Defaults

There are six types of default phrase, corresponding to the type of the object for which the defaults are being specified. The information specified within a defaults phrase is used in the derivation of attribute values within an ODA document. The defaults phrase corresponds to the attribute "Default Value Lists".

In the main, defaults phrases are composed of statements and phrases which also occur within the phrases describing objects and definitions. Since these statements and phrases are described within other sections, the descriptions are not repeated here. Only the exceptions to this rule are described in this section.

Within each type of default phrase, the FROM statement may be used to specify the name of the definition from which an object should be derived, in the absence of an explicitly specified name. It corresponds to the attribute "Reference to Object Definition".

Within the blo-defaults and block-defaults phrases, the PRESENTATION phrase permits the specification of default values for presentation attributes. The syntax in the default phrases differs from that in the object and definition phrases since it is possible to specify defaults for both character and photographic presentation attributes, whereas it is possible to have only a single content type within each object or definition.

5.13 Content Architecture

The content architecture statement specifies the options available with respect to content architecture within a particular basic object. The eight keyword options correspond directly to the content architectures defined in ECMA-101, 9.3.

5.14 Names

Those literals within the syntax the identifier of which ends in "-name" are classed as names. A name is a user-assigned identifier, and may be composed of upper and lower case characters, and hyphen.

5.15 Cardinal

A cardinal is an unsigned positive decimal number, composed of the digits 0-9.

5.15 Character Strings

A character string within a document script is composed of graphic characters and graphic representations of control functions, enclosed within quotation mark ("). Which particular characters are valid for any given character string is dependent upon the content architecture for that string. The representation of control functions is defined in Section 6 of this Technical Report. The representation of a quotation mark character within a character string is two quotation marks ("").

6. CONVENTIONS FOR CONTROL FUNCTIONS

This section defines the conventions to be used for the graphic representation of control functions within character strings in a document script.

Each control function defined within ODA is assigned a unique 2-character identifier. An instance of a particular control function within a character string is represented by this identifier bracketed by asterisks, Where the control function requires parameters, these follow the identifier, and are delimited from it, and from each other, by a single space character.

This convention can be described in a BNF style notation as follows:

"✱" <identifier>[""<parameter>]*"✱"

The control function identifiers and the legal parameter values are specified in Table 1.

ODA Standard Mnemonic	Specification Language Identifier	Legal Parameter Values
BS	BS	-
BPH	BP	-
CR	CR	-
DCS	DC	-
EM	EM	-
HPB	HB	cardinal
HPR	HR	cardinal
HT	HT	-
IGS	GS	identifier
LF	LF	-
NBH	NB	-
PLD	PD	-
PLU	PU	-
SGR	GR	BOLD, NORMAL, FEINT, ITALIC, NO-ITALIC, UNDERLINE, UNDERLINE-2, NO-UNDERLINE, STRIKEOUT, NO-STRIKEOUT, FONT-cardinal
SHS	HS	6, 10, 12, 15
SPI	SI	{cardinal, cardinal}
ST	ST	-
SUB	SB	-
SVS	VS	3, 4, 6, 8, 12

Table 1 - Control function identifiers and parameter values

7. COLLECTED SYNTAX

7.1 document

ODA DOCUMENT

LANGUAGE *language-version*

STANDARD *standard-version*

[FONT NAMES

{ *font-name* = *cardinal* } ...]

[MEASUREMENT UNITS

{ *unit-name* = *cardinal* [/ *cardinal*] { BMU / *unit-name* } } ...]

document-specification

7.2 document-specification

DOCUMENT *document-class-name* IS

[TITLE *character-string*]

[VERSION *character-string*]

[KEYWORDS { *character-string* } ...]

[REFERENCE *character-string*]

[DATED *date*]

[EXPIRES ON *date*]

[CLASSIFICATION *character-string*]

[LANGUAGES USED { *character-string* } ...]

[ORIGINATORS INFORMATION

[ORGANISATIONS { *character-string* } ...]

[PREPARED BY *character-string*]

[OWNED BY { *character-string* } ...]

[AUTHORS { *character-string* } ...]

[COPYRIGHT { *character-string* } ...]

[STATUS *character-string*]

[DISTRIBUTION { *character-string* } ...]

[OTHER INFORMATION { *character-string* } ...]]

[logical-definitions]

[layout-definitions]

[logical-structure]

[layout-structure]

[content-portions]

END DOCUMENT

7.3 logical-definitions

```
[ LOGICAL DEFINITIONS
    { logical-definition } ...
  END LOGICAL DEFINITIONS ]
```

7.4 logical-definition

```
definition-name DEFINES [ PROTECTED ] { DOCUMENT | COMPOSITE OBJECT | BASIC OBJECT }
  [ ANNOTATED WITH character-string ]
  [ COMPOSED OF construction ]
  [ [ { STARTING | CONTINUING } ] LAYOUT
    [ CATEGORISED BY category-name ]
    [ WITHIN NEW { definition-name | category-name } ]
    [ EXCLUSIVELY WITHIN definition-name ]
    [ UNDIVIDED WITHIN { definition-name | category-name } ]
    [ BOUNDED BY
      [ measure TOP ] [ measure BOTTOM ]
      [ measure LEFT ] [ measure RIGHT ] ]
    [ SEPARATED BY
      [ measure LEADING ] [ measure TRAILING ] ] ]
  [ DEFAULTS
    [ FOR COMPOSITE OBJECT clo-defaults ]
    [ FOR BASIC OBJECT blo-defaults ] ]
  [ LOCAL DATA { data-name = expression } ... ]
  [ CONTENT [ content-architecture ] [ { CHARACTER | PHOTOGRAPHIC } ]
    [ PRESENTATION { character-presentation-attributes | photographic-presentation-attributes } ]
    [ CONTAINING { character-string | string-expression | { content-portion-name } ... } ] ]
END definition-name
```

7.5 layout-definitions

```
[ LAYOUT DEFINITIONS
    { layout-definition } ...
  END LAYOUT DEFINITIONS ]
```

7.6 layout definition

```
definition-name DEFINES { DOCUMENT | PAGE-SET | PAGE | FRAME | BLOCK }
  [ ANNOTATED WITH character-string ]
  [ POSITIONED AT X measure Y measure ]
  [ WIDTH measure AND HEIGHT measure ]
  [ COMPOSED OF construction ]
  [ BLOCK PROGRESSION { LEFT | RIGHT | UP | DOWN } ]
  [ ACCEPTING CATEGORIES { category-name } ... ]
  [ BACKGROUND OPAQUE ]
  [ DEFAULTS
    [ FOR PAGE-SET page-set-defaults ]
    [ FOR PAGE page-defaults ]
    [ FOR FRAME frame-defaults ]
    [ FOR BLOCK block-defaults ] ]
  [ LOCAL DATA { data-name = expression } ... ]
  [ CONTENT [ content-architecture ] [ { CHARACTER | PHOTOGRAPHIC } ]
    [ PRESENTATION { character-presentation-attributes | photographic-presentation-attributes } ]
    [ CONTAINING { character-string | string-expression | { content-portion-name } ... } ] ]
END definition-name
```

7.7 <u>logical-structure</u>

logical-object

7.8 <u>logical-object</u>

```
                  ┌        LOGICAL DOCUMENT              ┐
[ PROTECTED ]    {                                        }  IS
                  └  [ { COMPOSITE } ]  OBJECT           ┘
                         { BASIC     }
  [ FROM definition-name ]
  [ ANNOTATED WITH character-string ]
  [ [ { STARTING   } ] LAYOUT
        { CONTINUING }
    [ CATEGORISED BY category-name ]
    [ WITHIN NEW { definition-name } ]
                 { category-name   }
    [ EXCLUSIVELY WITHIN definition-name ]
    [ UNDIVIDED WITHIN { definition-name } ]
                       { category-name   }
    [ BOUNDED BY
      [ measure TOP ] [ measure BOTTOM ]
      [ measure LEFT ] [ measure RIGHT ] ]
    [ SEPARATED BY
      [ measure LEADING ] [ measure TRAILING ] ] ]
  [ LOCAL DATA { data-name = expression }... ]
  [ SUBORDINATES
     { logical-object }... ]
  [ CONTENT [ content-architecture ] [ { CHARACTER    } ]
                                       { PHOTOGRAPHIC }
    [ PRESENTATION { character-presentation-attributes     } ]
                   { photographic-presentation-attributes  }
    [ CONTAINING {          character-string          } ] ]
                 { { content-portion-name }...        }
END
```

7.9 layout structure

layout-object

7.10 layout-object

```
{ LAYOUT DOCUMENT }
{      PAGE-SET   } IS
{      PAGE       }
{      FRAME      }
{      BLOCK      }
  [ FROM definition-name ]
  [ ANNOTATED WITH character-string ]
  [ POSITIONED AT X measure Y measure ]
  [ WIDTH measure AND HEIGHT measure ]
  [ BLOCK PROGRESSION { LEFT  } ]
                      { RIGHT }
                      { UP    }
                      { DOWN  } ]
  [ BACKGROUND OPAQUE ]
  [ LOCAL DATA { data-name = expression } ... ]
  [ SUBORDINATES
    { layout-object } ... ]
  [ CONTENT [ content-architecture ] [ { CHARACTER    } ]
                                     { PHOTOGRAPHIC } ]
    [ PRESENTATION { character-presentation-attributes    }
                   { photographic-presentation-attributes } ]
    [ CONTAINING { character-string                } ] ]
                 { { content-portion-name } ... }   ] ]
END
```

7.11 content-portions

```
[ CONTENT PORTIONS
    { content-portion } . . .
  END CONTENT PORTIONS ]
```

7.12 content-portion

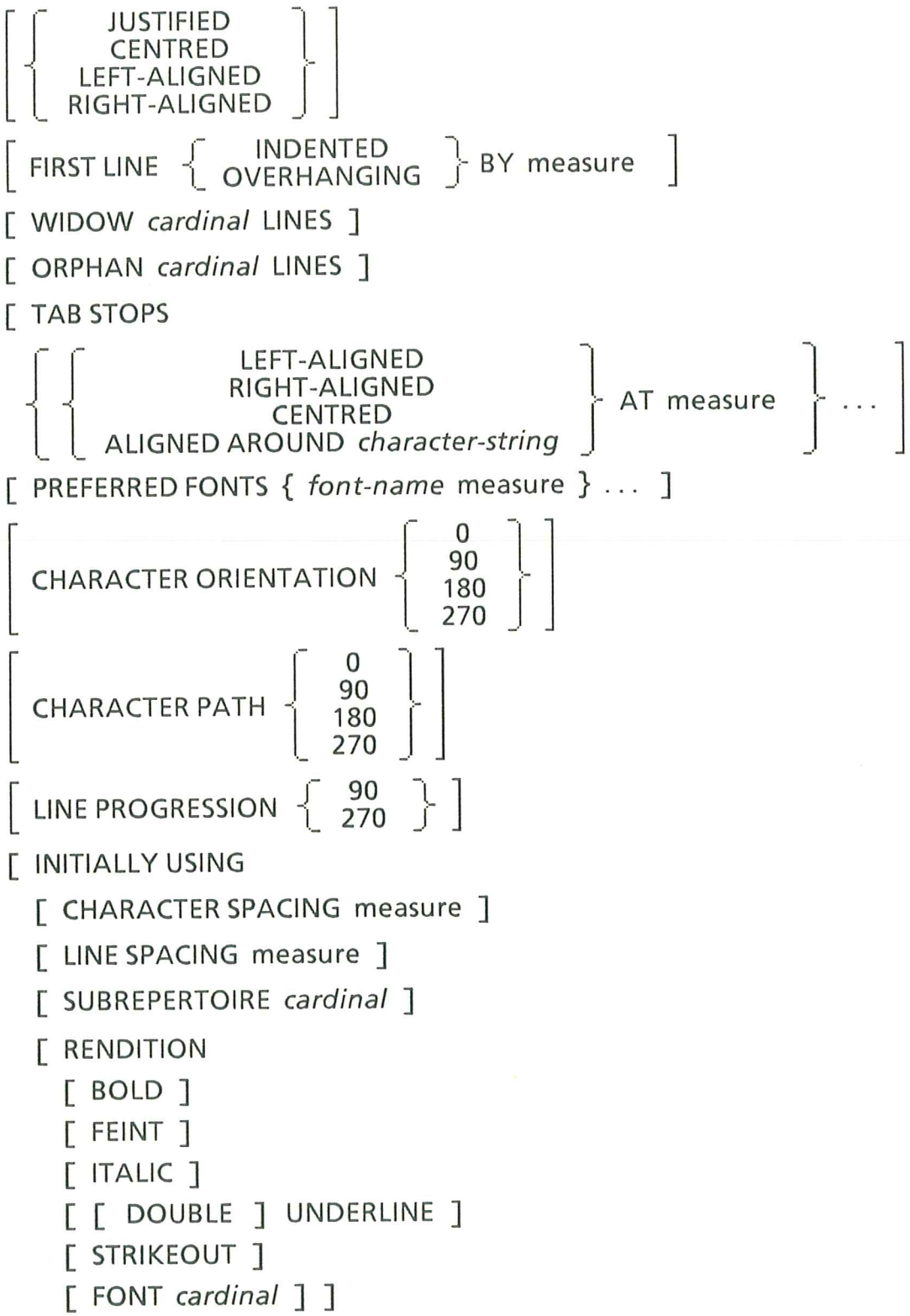
character content portion

```
content-portion-name IS CHARACTER
  { character-string }
  { FROM character-string }
END content-portion-name
```

photographic content portion

```
content-portion-name IS [ UNCOMPRESSED ] PHOTOGRAPHIC
  [ WITH
    [ cardinal PELS PER LINE ]
    [ cardinal PELS DISCARDED ]
    [ cardinal LINES ] ]
  FROM character-string
END content-portion-name
```

7.13 character-presentation-attributes



7.14 photographic-presentation-attributes

```
[ INITIAL OFFSET X measure Y measure ]
[ PEL PATH { 0   } ]
           { 90  }
           { 180 }
           { 270 }
[ LINE PROGRESSION { 90  } ]
                   { 270 }
[ { UNIFORM RESOLUTION cardinal                          } ]
  { RESOLUTION cardinal HORIZONTAL cardinal VERTICAL     }
```

7.15 content-architecture

```
{ CI-1 }
{ CI-2 }
{ CI-3 }
{ CP-0 }
{ CP-3 }
{ CF-3 }
{ PH-0 }
{ PH-1 }
```

7.16 measure

cardinal unit-name

7.17 expression

```
{ string-expression    }
{ cardinal-expression  }
{ object-id-expression }
```

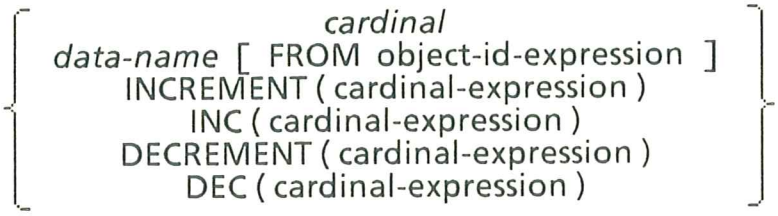
7.18 string-expression

{ atomic-string-expression } . . .

7.19 atomic-string-expression

```
{ character-string                                    }
{ data-name [ FROM object-id-expression ]             }
{ MAKE-STRING ( cardinal-expression )                 }
{ UPPER-ALPHA ( cardinal-expression )                 }
{ LOWER-ALPHA ( cardinal-expression )                 }
{ UPPER-ROMAN ( cardinal-expression )                 }
{ LOWER-ROMAN ( cardinal-expression )                 }
```

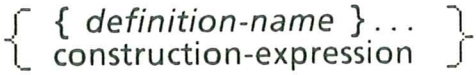
7.20 cardinal-expression



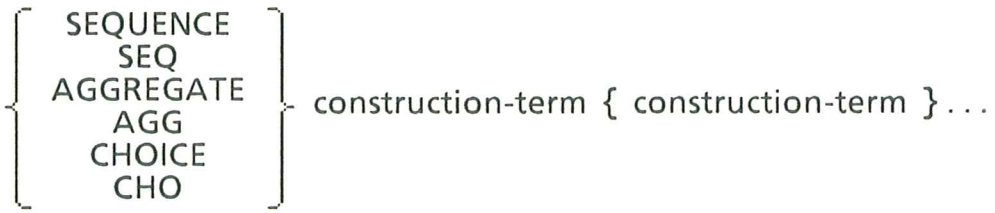
7.21 object-id-expression

7.22 object-selection-function

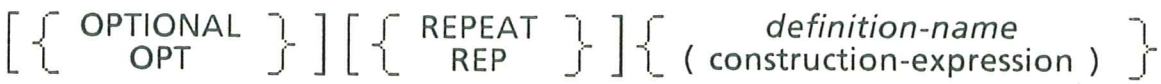
7.23 construction



7.24 construction-expression



7.25 construction-term



7.26 clo-defaults

[FROM *definition-name*]
[PROTECTED]
[LAYOUT
  [CATEGORISED BY *category-name*]
  [WITHIN NEW { *definition-name* / *category-name* }]
  [UNDIVIDED WITHIN { *definition-name* / *category-name* }]]

7.27 blo-defaults

[FROM *definition-name*]
[PROTECTED]
[[{ STARTING / CONTINUING }] LAYOUT
[CATEGORISED BY *category-name*]
[WITHIN NEW { *definition-name* / *category-name* }]
[UNDIVIDED WITHIN { *definition-name* / *category-name* }]
[BOUNDED BY
  [measure TOP] [measure BOTTOM]
  [measure LEFT] [measure RIGHT]]
[SEPARATED BY
  [measure LEADING] [measure TRAILING]]]
[PRESENTATION
  [FOR CHARACTER character-presentation-attributes]
  [FOR PHOTOGRAPHIC photographic-presentation-attributes]]

7.28 page-set-defaults

[FROM *definition-name*]

7.29 page-defaults

[FROM *definition-name*]
[WIDTH measure AND HEIGHT measure]

7.30 frame-defaults

[FROM *definition-name*]
[POSITIONED AT X measure Y measure]
[WIDTH measure AND HEIGHT measure]
[BLOCK PROGRESSION { LEFT | RIGHT | UP | DOWN }]
[ACCEPTING CATEGORIES { *category-name* } . . .]
[BACKGROUND OPAQUE]

7.31 block-defaults

[FROM *definition-name*]
[POSITIONED AT X measure Y measure]
[WIDTH measure AND HEIGHT measure]
[BACKGROUND OPAQUE]
[PRESENTATION
  [FOR CHARACTER character-presentation-attributes]
  [FOR PHOTOGRAPHIC photographic-presentation-attributes]]

APPENDIX A

GUIDELINES FOR PRESENTATION

This Appendix details the suggested guidelines for use in the composition of ODA documents using the specification language defined in this Technical Report. It is not mandatory that these guidelines be adhered to. They are presented here only in order to provide the basis for a common style of presentation for ODA documents in this form.

A.1 Order of Definitions, Objects and Content Portions

The syntax of the specification language defines a fixed order for many of the components of an ODA document. Where a unique order is not imposed, these guidelines provide a suggested order.

Within each set of definitions, the definitions should be listed in a depth first, left to right order, starting with the document root. Thus, for any construction expression, all the definitions used in the composition of each definition are specified before those used in the composition of subsequent definitions.

Within each composite object, those objects forming its composition should be listed in the order in which they are intended to appear in the corresponding ODA document structure.

Content portions should be listed in the order in which they appear in the remainder of the document specification.

A.2 Spacing and Indentation

Each object and definition should be separated from adjacent parts of the document specification, at both beginning and end, by additional vertical spacing, for example an extra blank line.

The composition of each composite object should be indented with respect to the remaining parts of the specification of that object. Indentation should also be used where a list of items follows an introductory keyword, for example in the clauses for originators information, defaults and presentation.

A.3 Location of Content

The specification language allows for character content to be embedded directly within an object or definition or to be located in a separate content portion which is then referred to from an object or definition. Wherever possible, content should be embedded directly. However, in the case where both logical and layout structures are specified, and hence where the same content belongs in more than one location, this content should be specified only once, within a separate content portion.

A.4 Format of Names

All names used within a document should be meaningful in the context of that document.

Where multiple-word names are used, the words should be separated by a hyphen. Each word in a name should have as initial letter a capital letter with the remainder of the word in small letters.

A.5 Use of Comments

The use of comments within the specification language is encouraged wherever these are likely to aid comprehension by a human reader. Comments should be neither cryptic nor overly verbose, and should never be used as a substitute for the use of meaningful names within a document.

APPENDIX B

EXAMPLES

This Appendix illustrates two examples of the use of the specification language to describe an ODA document as simply as possible. The first example is of the definition of a document class, whilst the second is an example of a document of this class.

The document specified in the examples is a report logically composed of a title, an abstract and a body, which contains paragraphs and images. Paragraphs may contain footnote references; footnotes must begin on the same page on which they are referenced; images are composed of a photograph and a caption.

The document is presented according to a layout organized in three types of page. The First-Page is composed of an upper frame of the full page width in which the Title and Abstract will be laid out; two equally dimensioned column frames, which occupy the most part of the rest of the page, in which the Body will be laid out; and a small page number block in the lower right hand corner. The subsequent pages will be either a full frame or a normal page containing two columns, depending upon the presence or absence of an Image on that page. In the first case, the page will be composed of an upper, full page width, frame which will contain the image, the caption and some body text; a Footnote frame to be filled with Footnote-Body; and a page number block identical to that on the first page. In the second case, the normal page replaces the full frame, with a left column frame and a right column frame, equally dimensioned, into which the paragraph text will be laid out.

The examples of document class and document are given in the following subsections, and are illustrated in Figures B.1 and B.2.

B.1 Example of a Document Class

```
ODA DOCUMENT

LANGUAGE 1.0
STANDARD 1.0

FONT NAMES
  Elitto = 1
MEASUREMENT UNITS
  mm = 12000/254 BMU

DOCUMENT Report-Class IS
  TITLE "Report-Class is an ODA-SL example"
  VERSION "1.0"
  KEYWORDS "Office Document Architecture" "Interchange Format"
    "TPF-2" "Class Definition"
  REFERENCE "ECMA/TC29/84/43"
  DATED 1984-06-01
  EXPIRES ON 2000-01-01
```

```
CLASSIFICATION "Contribution"
LANGUAGES USED "Computer English"
ORIGINATORS INFORMATION
  ORGANISATIONS "Xerox" "Olivetti" "ICL"
  PREPARED BY "Cooper" "Malinverno" "Robinson"
  OWNED BY "ECMA"
  AUTHORS "Cooper" "Malinverno" "Robinson"
  COPYRIGHT "ECMA for Europe"
  STATUS "Approved"
```

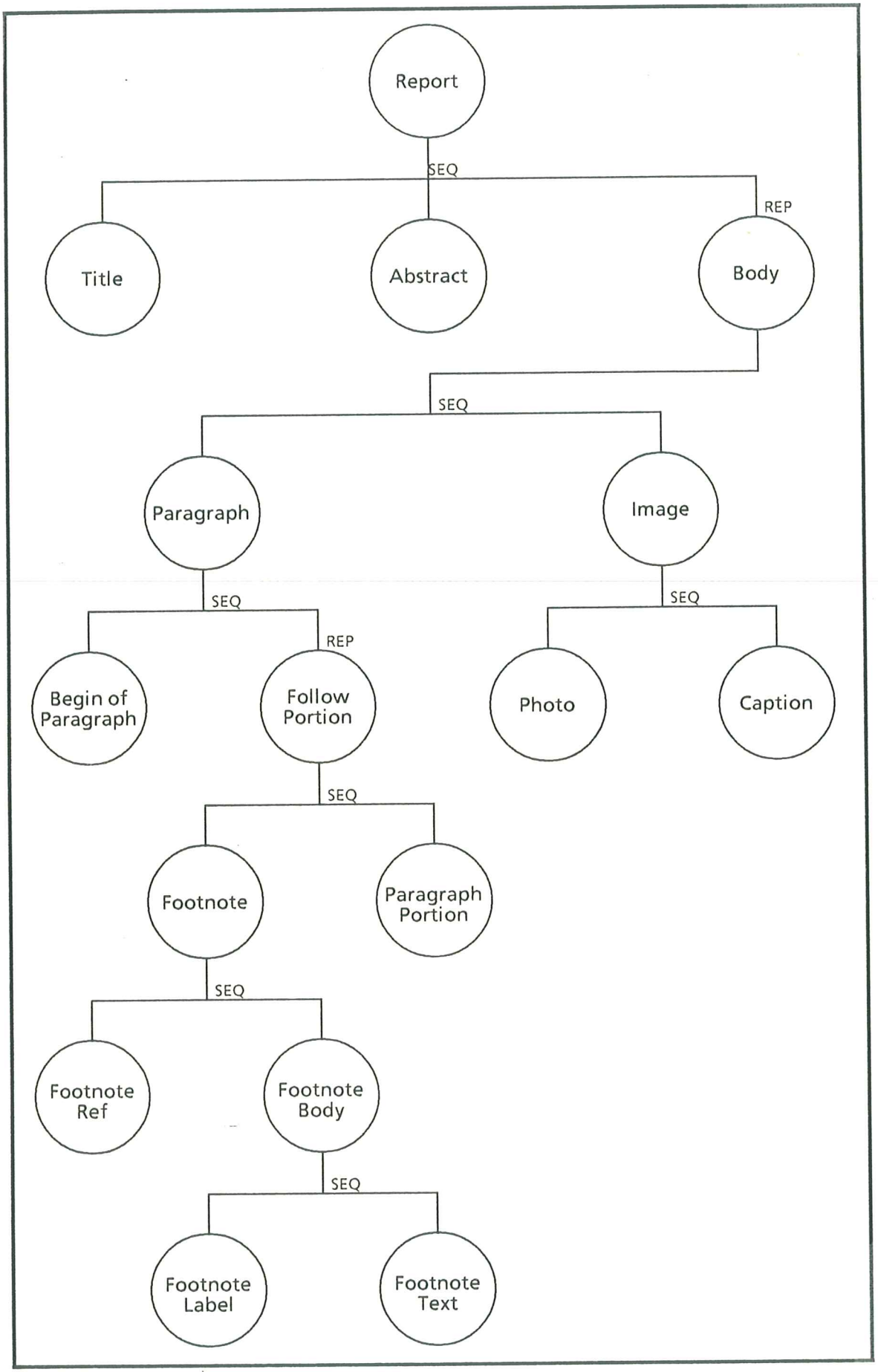


Figure B.1 - Logical definition of 'Report'

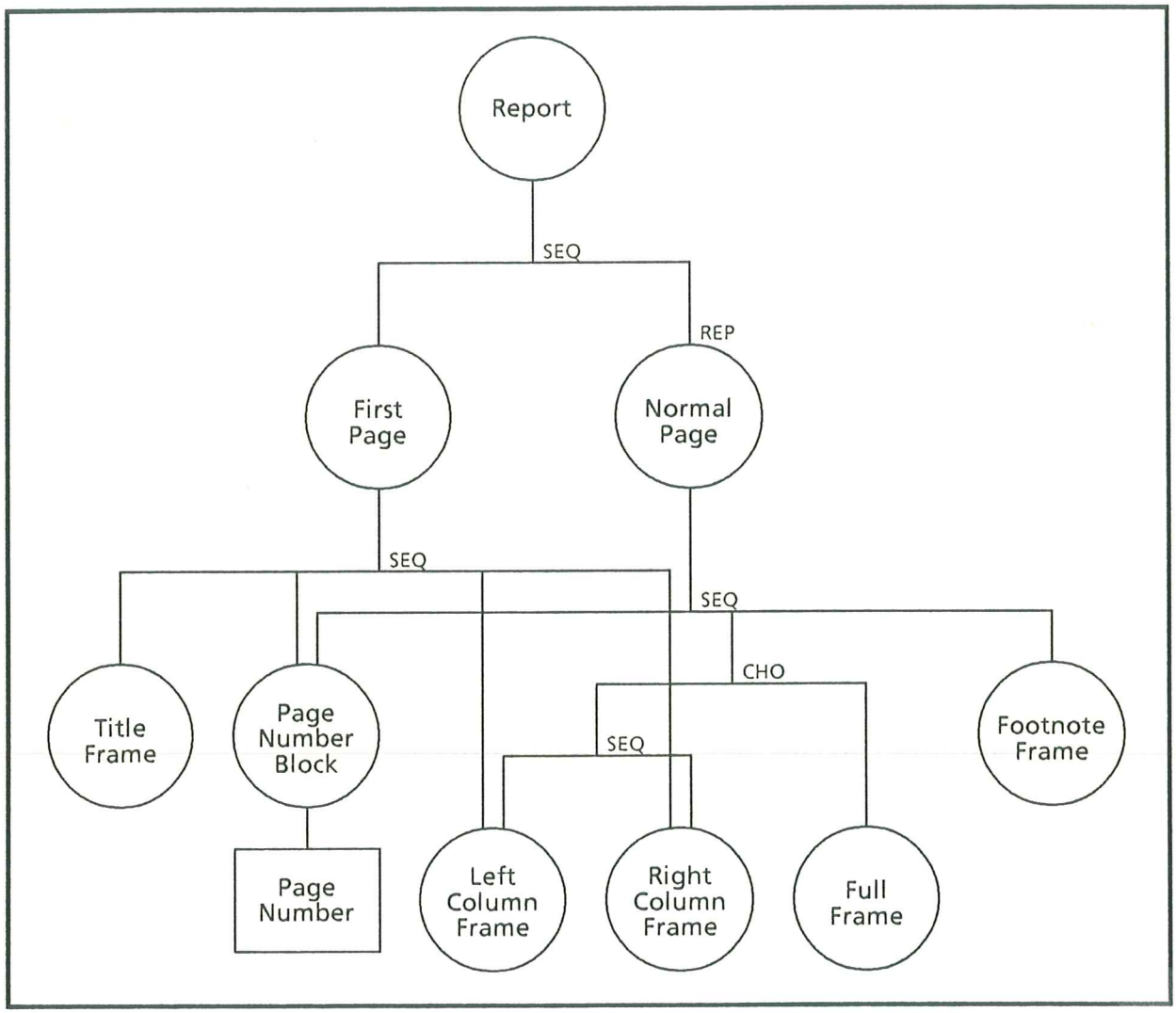


Figure B.2 - Layout definition of 'Report'

```
  DISTRIBUTION "All ECMA Members"
  OTHER INFORMATION "Internal Budget No. 91378"

LOGICAL DEFINITIONS

  Log-Report-Def DEFINES DOCUMENT
    COMPOSED OF Title-Def Abstract-Def REP Body-Def
    LAYOUT
      EXCLUSIVELY WITHIN Lay-Report-Def
  END Log-Report-Def

  Title-Def DEFINES BASIC OBJECT
    STARTING LAYOUT
      CATEGORISED BY Header
      WITHIN NEW Title-F-Def
    CONTENT CHARACTER
  END Title-Def
```

```
Abstract-Def DEFINES BASIC OBJECT
  STARTING LAYOUT
    CATEGORISED BY Header
  CONTENT CHARACTER
END Abstract

Body-Def DEFINES COMPOSITE OBJECT
  COMPOSED OF Paragraph-Def OPT Image-Def
END Body-Def

Paragraph-Def DEFINES COMPOSITE OBJECT
  COMPOSED OF
    Beginning-of-Paragraph-Def OPT REP Follow-Portion-Def
  DEFAULTS
    FOR BASIC OBJECT
      PRESENTATION
        JUSTIFIED
        FIRST LINE INDENTED BY 15 mm
END Paragraph-Def

Beginning-of-Paragraph-Def DEFINES BASIC OBJECT
  STARTING LAYOUT
  CONTENT CHARACTER
END Beginning-of-Paragraph-Def

Follow-Portion-Def DEFINES COMPOSITE OBJECT
  COMPOSED OF Footnote-Def OPT Paragraph-Portion-Def
END Follow-Portion-Def

Footnote-Def DEFINES COMPOSITE OBJECT
  COMPOSED OF Ref-to-Footnote-Def Footnote-Body-Def
  LAYOUT
    UNDIVIDED WITHIN Normal-Page-Def
END Footnote-Def

Ref-to-Footnote-Def DEFINES BASIC OBJECT
  CONTENT CHARACTER
    CONTAINING "┼"
END Ref-to-Footnote-Def

Footnote-Body-Def DEFINES COMPOSITE OBJECT
  COMPOSED OF Label-Def Text-Def
  LAYOUT
    CATEGORISED BY Footnote
  DEFAULTS
    FOR BASIC OBJECT
      PRESENTATION
        FIRST LINE OVERHANGING BY 8 mm
END Footnote-Body-Def

Label-Def DEFINES BASIC OBJECT
  STARTING LAYOUT
  CONTENT CHARACTER
    CONTAINING "┼"*CR*"
END Label-Def
```

```
  Text-Def DEFINES BASIC OBJECT
    CONTINUING LAYOUT
    CONTENT CHARACTER
  END Text-Def

  Paragraph-Portion-Def DEFINES BASIC OBJECT
    CONTENT CHARACTER
  END Paragraph-Portion-Def

  Image-Def DEFINES COMPOSITE OBJECT
    COMPOSED OF Photo-Def Caption-Def
    LAYOUT
      UNDIVIDED WITHIN Full-F-Def
  END Image-Def

  Photo-Def DEFINES BASIC OBJECT
    CONTENT PHOTOGRAPHIC
  END Photo-Def

  Caption-Def DEFINES BASIC OBJECT
    CONTENT CHARACTER
      PRESENTATION
        CENTRED
  END Caption-Def

END LOGICAL DEFINITIONS

LAYOUT DEFINITIONS

  Lay-Report-Def DEFINES DOCUMENT
    COMPOSED OF First-Page-Def REP Normal-Page-Def
  END Lay-Report-Def

  First-Page-Def DEFINES PAGE
    WIDTH 210 mm AND HEIGHT 297 mm  /*A4 dimensions*/
    COMPOSED OF
      Title-F-Def Page-Number-B L-Column-Def R-Column-Def
    LOCAL DATA
      Page-Number = 1
      Col-Y = 80 mm
      Col-H = 177 mm
  END First-Page-Def

  Title-F-Def DEFINES FRAME
    POSITIONED AT X 15 mm Y 15 mm
    WIDTH 180 mm AND HEIGHT 60 mm
    ACCEPTING CATEGORIES Header
    DEFAULTS
      FOR BLOCK
        PRESENTATION
          FOR CHARACTER
            PREFERRED FONTS Elitto 4 mm
            RENDITION BOLD
  END Title-F-Def

  Page-Number-B DEFINES BLOCK
    POSITIONED AT X 180 mm Y 267 mm
```

```
      WIDTH 20 mm AND HEIGHT 10 mm
      CONTENT CHARACTER
      CONTAINING MAKE-STRING(Page-Number FROM SUP)
    END Page-Number-B

    L-Column-Def DEFINES FRAME
      POSITIONED AT X 15 mm Y Col-Y FROM SUP
      WIDTH 85 mm AND HEIGHT  Col-H FROM SUP
      DEFAULTS
        FOR BLOCK
          PRESENTATION
            FOR CHARACTER
              JUSTIFIED
              PREFERRED FONTS Elitto 4 mm
    END L-Column-Def

    R-Column-Def DEFINES FRAME
      POSITIONED AT X 110 mm Y Col-Y FROM SUP
      WIDTH 85 mm AND HEIGHT Col-H FROM SUP
      DEFAULTS
        FOR BLOCK
          PRESENTATION
            FOR CHARACTER
              JUSTIFIED
              PREFERRED FONTS Elitto 4 mm
    END R-Column-Def

    Normal-Page-Def DEFINES PAGE
      WIDTH 210 mm AND HEIGHT 297 mm   /*A4 dimensions*/
      COMPOSED OF
        Footnote-F-Def Page-Number-B
        (CHO(L-Col-Def R-Col-Def) Full-F-Def)
      LOCAL DATA
        Page-Number = INC (Page-Number FROM PREC)
        Col-Y = 15 mm
        Col-H = 212 mm
    END Normal-Page-Def

    Footnote-F-Def DEFINES FRAME
      POSITIONED AT X 15 mm Y 237 mm
      WIDTH 180 mm AND HEIGHT 20 mm
      ACCEPTING CATEGORIES Footnote
        CONTENT CHARACTER
          PRESENTATION
            JUSTIFIED
            PREFERRED FONTS Elitto 4 mm
            RENDITION ITALIC
    END Footnote-F-Def

    Full-F-Def DEFINES FRAME
      POSITIONED AT X 15 mm Y 15 mm
      WIDTH 180 mm AND HEIGHT 121 mm
    END Full-F-Def

  END LAYOUT DEFINITIONS

END DOCUMENT
```

B.2 Example of a Document Instance

```
ODA DOCUMENT

LANGUAGE 1.0
STANDARD 1.0

FONT NAMES
  Elitto = 1
MEASUREMENT UNITS
  mm = 12000/254 BMU

DOCUMENT Report IS
  TITLE "Report is an ODA-SL example"
  VERSION "1.0"
  KEYWORDS "Office Document Architecture" "Interchange Format"
    "MPF-1" "Document Instance"
  REFERENCE "ECMA/TC29/84/43"
  DATED 1984-06-01
  EXPIRES ON 2000-01-01
  CLASSIFICATION "Contribution"
  LANGUAGES USED "Computer English"
  ORIGINATORS INFORMATION
    ORGANISATIONS "Xerox" "Olivetti" "ICL"
    PREPARED BY "Cooper" "Malinverno" "Robinson"
    OWNED BY "ECMA"
    AUTHORS "Cooper" "Malinverno" "Robinson"
    COPYRIGHT "ECMA for Europe"
    STATUS "Approved"
    DISTRIBUTION "All ECMA Members"
    OTHER INFORMATION "Internal Budget No. 91378"

  LOGICAL DOCUMENT IS
    FROM Log-Report-Def
    LAYOUT
      EXCLUSIVELY WITHIN Lay-Report-Def
    SUBORDINATES

      BASIC OBJECT IS
        FROM Title-Def
        STARTING LAYOUT
          CATEGORISED BY Header
          WITHIN NEW Title-F-Def
        CONTENT CHARACTER
          CONTAINING Title-Cont
      END

      BASIC OBJECT IS
        FROM Abstract-Def
        STARTING LAYOUT
          CATEGORISED BY Header
        CONTENT CHARACTER
          CONTAINING Abstract-Cont
      END
```

```
COMPOSITE OBJECT IS
  FROM Body-Def
  SUBORDINATES

COMPOSITE OBJECT IS
  FROM Paragraph-Def
  SUBORDINATES

    BASIC OBJECT IS
      FROM Beginning-of-Paragraph-Def
      STARTING LAYOUT
      CONTENT CHARACTER
        CONTAINING Begin-Para-Content

    END

    COMPOSITE OBJECT IS
      FROM Follow-Portion-Def
      SUBORDINATES

        COMPOSITE OBJECT IS
          FROM Footnote-Def
          UNDIVIDED WITHIN Normal-Page-Def
          SUBORDINATES

            BASIC OBJECT IS
              FROM Ref-to-Footnote-Def
            END

            COMPOSITE OBJECT IS
              FROM Footnote-Body-Def
              LAYOUT
                EXCLUSIVELY WITHIN
                  Footnote-F-Def
              SUBORDINATES

                BASIC OBJECT IS
                  FROM Label-Def
                END

                BASIC OBJECT IS
                  FROM Text-Def
                  CONTENT CHARACTER
                    CONTAINING Text-Cont

                END

            END

        END

        BASIC OBJECT IS
          FROM Paragraph-Portion-Def
          CONTENT CHARACTER
```

```
            CONTAINING Para-Cont

          END

      END

    END

    COMPOSITE OBJECT IS
      FROM Image-Def
      LAYOUT
        UNDIVIDED WITHIN Full-F-Def
      SUBORDINATES

          BASIC OBJECT IS
            FROM Photo-Def
            STARTING LAYOUT
            CONTENT PHOTOGRAPHIC
              CONTAINING Photo-Cont

          END

          BASIC OBJECT IS
            FROM Caption-Def
            CONTENT CHARACTER
              PRESENTATION
                CENTRED
              CONTAINING Caption-Cont

          END

      END

    END

END

LAYOUT DOCUMENT IS
  FROM Lay-Report-Def
  SUBORDINATES

    PAGE IS
      FROM First-Page-Def
      WIDTH 210 mm AND HEIGHT 297 mm
      SUBORDINATES

        FRAME IS
          FROM Title-F-Def
          POSITIONED AT X 15 mm Y 15 mm
            WIDTH 180 mm AND HEIGHT 60 mm
          ACCEPTING CATEGORIES Header
          SUBORDINATES

            BLOCK IS
              POSITIONED AT X 15 mm Y 15 mm
              WIDTH 180 mm AND HEIGHT 10 mm
              CONTENT CHARACTER
```

```
            CONTAINING Abstract-Cont

        END

        BLOCK IS
          POSITIONED AT X 15 mm Y 25 mm
          WIDTH 180 mm AND HEIGHT 50 mm
          CONTENT CHARACTER
            CONTAINING Abstract-Cont

        END
    END

    BLOCK IS
      POSITIONED AT X 180 mm Y 267 mm
      WIDTH 20 mm AND HEIGHT 10 mm
      CONTENT CHARACTER
        CONTAINING "1"

    END

    FRAME IS
      FROM L-Column-Def
      POSITIONED AT X 15 mm Y 80 mm
      WIDTH 85 mm AND HEIGHT 177 mm
      SUBORDINATES

      BLOCK IS
        CONTENT CHARACTER
          CONTAINING LF-Para-Cont

      END

  END

  FRAME IS
    FROM R-Column-Def
    POSITIONED AT X 110 mm Y 80 mm
    WIDTH 85 mm AND HEIGHT 177 mm
    SUBORDINATES

      BLOCK IS
        CONTENT CHARACTER
          CONTAINING RF-Para-Cont

      END

  END

END

PAGE IS
  FROM Normal-Page-Def
  WIDTH 210 mm AND HEIGHT 297 mm
  SUBORDINATES
```

```
    BLOCK IS
      POSITIONED AT X 180 mm Y 267 mm
      WIDTH 20 mm AND HEIGHT 10 mm
      CONTENT CHARACTER
        CONTAINING "2"

    END

    FRAME IS
      FROM Full-F-Def
      WIDTH 180 mm AND HEIGHT 212 mm
      SUBORDINATES

        BLOCK IS
          WIDTH 180 mm AND HEIGHT 50 mm
          CONTENT PHOTOGRAPHIC
            CONTAINING Photo-Cont

        END

        BLOCK IS
          POSITIONED AT X 15 mm Y 50 mm
          WIDTH 180 mm AND HEIGHT 10 mm
          CONTENT CHARACTER
            CONTAINING Caption-Cont

        END

        BLOCK IS
          POSITIONED AT X 15 mm Y 60 mm
          WIDTH 180 mm AND HEIGHT 152 mm
          CONTENT CHARACTER
            CONTAINING Text-Cont

        END

    END

    FRAME IS
      FROM Footnote-F-Def
      POSITIONED AT X 15 mm Y 237 mm
      WIDTH 180 mm AND HEIGHT 20 mm
      SUBORDINATES

        BLOCK IS
          CONTENT CHARACTER
            CONTAINING Note-Cont-1

        END

    END

END

PAGE IS
  FROM Normal-Page-Def
  WIDTH 210 mm AND HEIGHT 297 mm
```

```
    SUBORDINATES

      BLOCK IS
        POSITIONED AT X 180 mm Y 267 mm
        WIDTH 20 mm AND HEIGHT 10 mm
        CONTENT CHARACTER
          CONTAINING "3"

      END

      FRAME IS
        FROM L-Column-Def
        POSITIONED AT X 15 mm Y 15 mm
        WIDTH 85 mm AND HEIGHT 242 mm
        SUBORDINATES

          BLOCK IS
            CONTENT CHARACTER
              CONTAINING Col-Cont

          END

      END

      FRAME IS
        FROM R-Column-Def
        POSITIONED AT X 110 mm Y 15 mm
        WIDTH 85 mm AND HEIGHT 242 mm
        SUBORDINATES

          BLOCK IS
            CONTENT CHARACTER
              CONTAINING R-Col-Cont

          END

      END

      FRAME IS
        FROM Footnote-F-Def
        POSITIONED AT X 15 mm Y 237 mm
        WIDTH 180 mm AND HEIGHT 20 mm
        SUBORDINATES

          BLOCK IS
            WIDTH 180 mm AND HEIGHT 16 mm
            CONTENT CHARACTER
              CONTAINING Note-Cont-2

          END

      END

  END

END
```

```
CONTENT PORTIONS

  Title-Cont IS CHARACTER
     "Example Report"
  END Title-Cont

  Abstract-Cont IS CHARACTER
     "..."
  END Abstract-Cont

  Begin-Para-Cont IS CHARACTER
     "..."
  END Begin-Para-Cont

  Para-Cont IS CHARACTER
     "..."
  END Para-Cont

  Text-Cont IS CHARACTER
     "..."
  END Text-Cont

  Photo-Cont IS PHOTOGRAPHIC
     WITH
       2440 PELS PER LINE
       0 PELS DISCARDED
       400 LINES
     FROM "Example.img"
  END Photo-Cont

  Caption-Cont IS CHARACTER
     "..."
  END Caption-Cont

END CONTENT PORTIONS

END DOCUMENT
```

APPENDIX C

LIST OF KEYWORDS

ACCEPTING
AGG
AGGREGATE
ALIGNED
AND
ANNOTATED
AROUND
AT
AUTHORS
BACKGROUND
BASIC
BLOCK
BMU
BOLD
BOTTOM
BOUNDED
BY
CATEGORIES
CATEGORISED
CENTRED
CF-3
CHARACTER
CHO
CHOICE
CI-1
CI-2
CI-3
CLASSIFICATION
COMPOSED
COMPOSITE
CONTAINING
CONTENT
CONTINUING
COPYRIGHT
CP-0
CP-3
CURRENT-ID
DATA
DATED
DEC
DECREMENT
DEFAULTS
DEFINES
DEFINITIONS
DISCARDED
DISTRIBUTION
FOR
FRAME
FROM
HEIGHT
HORIZONTAL
INC
INCREMENT
INDENTED
INFORMATION
INITIAL
INITIALLY
IS
ITALIC
JUSTIFIED
KEYWORDS
LANGUAGE
LANGUAGES
LAYOUT
LEADING
LEFT
LEFT-ALIGNED
LINE
LINES
LOCAL
LOGICAL
LOWER-ALPHA
LOWER-ROMAN
MAKE-STRING
MEASUREMENT
NAMES
NEW
OBJECT
ODA
OF
OFFSET
ON
OPAQUE
OPT
OPTIONAL
ORGANIZATIONS
ORIENTATION
ORIGINATORS
ORPHAN
OTHER
OVERHANGING
OWNED
POSITIONED
PREC
PRECEDING
PREFERRED
PREPARED
PRESENTATION
PROGRESSION
PROTECTED
REFERENCE
RENDITION
REP
REPEAT
RESOLUTION
RIGHT
RIGHT-ALIGNED
SEPARATED
SEQ
SEQUENCE
SPACING
STANDARD
STARTING
STATUS
STOPS
STRIKE-OUT
SUBORDINATES
SUBREPERTOIRE
SUCC
SUCCEEDING
SUP
SUPERIOR
TAB
TITLE
TOP
TRAILING
UNCOMPRESSED
UNDERLINE
UNDIVIDED
UNIFORM
UNITS
UP
UPPER-ALPHA
UPPER-ROMAN
USED
USING
VERSION
VERTICAL

DOCUMENT
DOUBLE
DOWN
END
EXCLUSIVELY
EXPIRES
FEINT
FIRST
FONT
FONTS
PAGE
PAGE-SET
PATH
PEL
PELS
PER
PH-0
PH-1
PHOTOGRAPHIC
PORTIONS
WIDOW
WIDTH
WITH
WITHIN
X
Y