Contents

Page

1 Scope 1
2 Conformance 2
3 Normative References 3
4 Definitions 4
5 Notation and Conventions 7
5.1 Syntax Productions 7
5.2 Procedure Prototypes 8
6 DSSSL Overview 9
6.1 Areas of Standardization 9
6.2 Conceptual Model 10
6.3 DSSSL Languages 11
6.3.1 The Transformation Language 11
6.3.1.1 Components of the Transformation Process 12
6.3.1.2 Model for Coded Characters, Characters, and Glyph Identifiers 13
6.3.2 The Style Language 15
6.3.2.1 Components of the Formatting Process 15
6.3.2.2 Grove Building 16
6.3.2.3 Flow Object Tree 16
6.3.2.4 Flow Object Classes 17
6.3.2.5 Areas 18
6.3.2.6 Page and Column Geometry 18
6.3.2.7 Expression Language 19
6.3.2.8 Model for Coded Characters, Characters, and Glyph Identifiers 19
7 DSSSL Specifications 20
7.1 DSSSL Document Architecture 20
7.1.1 Features 25
7.1.2 SGML Grove Plan 25
7.1.3 Character Repertoire 25
7.1.4 Standard Characters 25
7.1.5 Other Characters 26
7.1.6 Baseset Encoding 26
7.1.7 Literal Described Character 27
7.1.8 Sdata Entity Mapping 27
7.1.9 Separator Characters 27
7.1.10 Name Characters 28
7.1.11 Character Combination 28
7.2 Public Identifiers 28
7.3 Lexical Conventions 28
7.3.1 Case Sensitivity 28

© ISO/IEC 1996
All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

7.3.2 Identifiers 28
7.3.3 Tokens, Whitespace, and Comments 29
8 Expression Language 29
8.1 Overview of the Expression Language 30
8.2 Basic Concepts 31
8.2.1 Variables and Regions 31
8.2.2 True and False 32
8.2.3 External Representations 32
8.2.4 Disjointness of Types 32
8.3 Expressions 32
8.3.1 Primitive Expression Types 33
8.3.1.1 Variable Reference 33
8.3.1.2 Literals 33
8.3.1.3 Procedure Call 34
8.3.1.4 Lambda Expression 35
8.3.1.5 Conditional Expression 36
8.3.2 Derived Expression Types 37
8.3.2.1 Cond-expression 37
8.3.2.2 Case-expression 37
8.3.2.3 And-expression 38
8.3.2.4 Or-expression 38
8.3.2.5 Binding expressions 38
8.3.2.6 Named-let 40
8.3.2.7 Quasiquotation 40
8.4 Definitions 42
8.5 Standard Procedures 43
8.5.1 Booleans 44
8.5.1.1 Negation 44
8.5.1.2 Boolean Type Predicate 44
8.5.2 Equivalence 45
8.5.3 Pairs and Lists 45
8.5.3.1 Pair Type Predicate 47
8.5.3.2 Pair Construction Procedure 47
8.5.3.3 car Procedure 47
8.5.3.4 cdr Procedure 47
8.5.3.5 c…r Procedures 48
8.5.3.6 Empty List Type Predicate 48
8.5.3.7 List Type Predicate 49
8.5.3.8 List Construction 49
8.5.3.9 List Length 49
8.5.3.10 Lists Appendance 49
8.5.3.11 List Reversal 50
8.5.3.12 Sublist Extraction 50
8.5.3.13 List Access 50
8.5.3.14 List Membership 50
8.5.3.15 Association Lists 51
8.5.4 Symbols 51
8.5.4.1 Symbol Type Predicate 51

8.5.4.2 Symbol to String Conversion 51
8.5.4.3 String to Symbol Conversion 52
8.5.5 Keywords.. 52
8.5.5.1 Keyword Type Predicate 52
8.5.5.2 Keyword to String Conversion.. 52
8.5.5.3 String to Keyword Conversion.. 53
8.5.6 Named Constants.. 53
8.5.7 Quantities and Numbers 53
8.5.7.1 Numerical Types... 53
8.5.7.2 Exactness 54
8.5.7.3 Implementation Restrictions.. 54
8.5.7.4 Syntax of Numerical Constants 55
8.5.7.5 Number Type Predicates 57
8.5.7.6 Exactness Predicates.. 57
8.5.7.7 Comparison Predicates 57
8.5.7.8 Numerical Property Predicates.. 58
8.5.7.9 Maximum and Minimum.. 58
8.5.7.10 Addition 58
8.5.7.11 Multiplication 58
8.5.7.12 Subtraction... 59
8.5.7.13 Division 59
8.5.7.14 Absolute Value 59
8.5.7.15 Number-theoretic Division.. 59
8.5.7.16 Real to Integer Conversion.. 60
8.5.7.17 e^n and Natural Logarithm 61
8.5.7.18 Trigonometric Functions 61
8.5.7.19 Inverse Trigonometric Functions.. 61
8.5.7.20 Square Root 61
8.5.7.21 Exponentiation.. 61
8.5.7.22 Exactness Conversion.. 62
8.5.7.23 Quantity to Number Conversion.. 62
8.5.7.24 Number to String Conversion.. 62
8.5.7.25 String to Number Conversion.. 63
8.5.8 Characters 63
8.5.8.1 Character Properties 64
8.5.8.2 Language-dependent Operations 65
8.5.8.3 Character Type Predicate.. 68
8.5.8.4 Character Comparison Predicates.. 68
8.5.8.5 Case-insensitive Character Predicates 68
8.5.8.6 Character Case Conversion 68
8.5.8.7 Character Properties 68
8.5.9 Strings... 69
8.5.9.1 String Type Predicate 69
8.5.9.2 String Construction... 69
8.5.9.3 String Length 70
8.5.9.4 String Access 70
8.5.9.5 String Equivalence.. 70
8.5.9.6 String Comparison.. 70

8.5.9.7 Substring Extraction....70
8.5.9.8 String Appendance....71
8.5.9.9 Conversion between Strings and Lists....71
8.5.10 Procedures....71
8.5.10.1 Procedure Type Predicate....71
8.5.10.2 Procedure Application....71
8.5.10.3 Mapping Procedures over Lists....72
8.5.10.4 External Procedures....72
8.5.11 Date and Time....72
8.5.12 Error Signaling....73
8.6 Core Expression Language....73
8.6.1 Syntax....73
8.6.2 Procedures....74
9 Groves....76
9.1 Nodal Properties....76
9.2 Grove Plans....78
9.3 Property Set Definition....79
9.3.1 Common Attributes....79
9.3.1.1 Component Names....79
9.3.1.2 Specification Documents....80
9.3.2 Modules....80
9.3.3 Data Type Definition....80
9.3.4 Class Definition....82
9.3.5 Property Definition....82
9.3.6 Normalization Rule Definition....83
9.4 Intrinsic Properties....83
9.5 Auxiliary Groves....85
9.6 SGML Property Set....85
9.7 DSSSL SGML Grove Plan....123
10 Standard Document Query Language....123
10.1 Primitive Procedures....124
10.1.1 Application Binding....124
10.1.2 Node Lists....124
10.1.3 Named Node Lists....125
10.1.4 Error Reporting....125
10.1.5 Application Name Transformation....126
10.1.6 Property Values....126
10.1.7 SGML Grove Construction....127
10.2 Derived Procedures....127
10.2.1 HyTime Support....127
10.2.2 List Operations....131
10.2.3 Generic Property Operations....137
10.2.4 Core Query Language....143
10.2.4.1 Navigation....144
10.2.4.2 Counting....144
10.2.4.3 Accessing Attribute Values....145
10.2.4.4 Testing Current Location....146
10.2.4.5 Entities and Notations....146

10.2.4.6 Name Normalization.. 148
10.2.5 SGML Property Operations.. 148
10.3 Auxiliary Parsing.. 150
10.3.1 Word Searching 150
10.3.2 Node Regular Expressions... 151
10.3.3 Regexp Constructors... 151
10.3.4 Regular Expression Searching Procedures 153
11 Transformation Language... 153
11.1 Features.. 154
11.2 Associations.. 154
11.3 Transform-expression 155
11.3.1 Subgrove-spec 155
11.3.2 Create-spec 157
11.3.3 Result-node-list... 158
11.3.4 Transform-grove-spec 159
11.3.5 SGML Prolog Parsing 160
11.4 SGML Document Generator 160
11.4.1 Verification Mapping 160
11.4.2 Transliteration... 162
12 Style Language 162
12.1 Features.. 163
12.2 Flow Object Tree 165
12.3 Areas... 165
12.3.1 Display Areas 166
12.3.2 Inline Areas.. 169
12.3.3 Inlined and Displayed Flow Objects 172
12.3.4 Attachment Areas 173
12.4 Flow Object Tree Construction 174
12.4.1 Construction Rules 174
12.4.2 Primary Flow Object 177
12.4.3 Sosofos.. 177
12.4.4 Multi-process Feature 181
12.4.5 Styles 181
12.4.6 Characteristic Specification.. 182
12.4.7 Synchronization of Flow Objects 185
12.5 Common Data Types and Procedures 186
12.5.1 Layout-driven Generated Text.. 186
12.5.1.1 Constructing Indirect Sosofos 187
12.5.1.2 Layout Numbering... 188
12.5.1.3 Reference Values 189
12.5.2 Length Specification... 191
12.5.3 Decoration Areas 191
12.5.4 Spaces 192
12.5.4.1 Display Spaces... 192
12.5.4.2 Inline Spaces.. 192
12.5.5 Glyph Identifiers... 193
12.5.6 Glyph Substitution Tables 193
12.5.7 Font Information... 194

12.5.8 Addresses 195
12.5.9 Color 196
12.6 Flow Object Classes 198
12.6.1 Sequence Flow Object Class 198
12.6.2 Display-group Flow Object 198
12.6.3 Simple-page-sequence Flow Object Class 200
12.6.4 Page-sequence Flow Object Class 202
12.6.4.1 Page-model 203
12.6.5 Column-set-sequence Flow Object Class 206
12.6.5.1 Column-set-model 208
12.6.6 Paragraph Flow Object Class 218
12.6.6.1 Line Spacing 226
12.6.7 Paragraph-break Flow Object Class 226
12.6.8 Line-field Flow Object Class 226
12.6.9 Sideline Flow Object Class 227
12.6.10 Anchor Flow Object Class 228
12.6.11 Character Flow Object Class 229
12.6.11.1 Character Properties 235
12.6.12 Leader Flow Object Class 237
12.6.13 Embedded-text Flow Object Class 238
12.6.14 Rule Flow Object Class 239
12.6.15 External-graphic Flow Object Class 243
12.6.16 Included-container-area Flow Object Class 248
12.6.17 Score Flow Object Class 252
12.6.18 Box Flow Object Class 254
12.6.19 Side-by-side Flow Object Class 259
12.6.20 Side-by-side-item Flow Object Class 261
12.6.21 Glyph-annotation Flow Object Class 262
12.6.22 Alignment-point Flow Object Class 263
12.6.23 Aligned-column Flow Object Class 263
12.6.24 Multi-line-inline-note Flow Object Class 266
12.6.25 Emphasizing-Mark Flow Object Class 267
12.6.26 Flow Object Classes for Mathematical Formulae 268
12.6.26.1 Math-sequence Flow Object Class 268
12.6.26.2 Unmath Flow Object Class 269
12.6.26.3 Subscript Flow Object Class 270
12.6.26.4 Superscript Flow Object Class 270
12.6.26.5 Script Flow Object Class 270
12.6.26.6 Mark Flow Object Class 272
12.6.26.7 Fence Flow Object Class 273
12.6.26.8 Fraction Flow Object Class 273
12.6.26.9 Radical Flow Object Class 274
12.6.26.10 Math-operator Flow Object Class 275
12.6.26.11 Grid Flow Object Class 276
12.6.26.12 Grid-cell Flow Object Class 277
12.6.27 Flow Object Classes for Tables 277
12.6.27.1 Table Flow Object Class 278
12.6.27.2 Table-part Flow Object Class 281

12.6.27.3 Table-column flow object.. 283
12.6.27.4 Automatic Table-width Computation.................................. 285
12.6.27.5 Table-row Flow Object Class.. 285
12.6.27.6 Table-cell Flow Object Class 285
12.6.27.7 Table-border Flow Object Class.. 288
12.6.28 Flow Object Classes for Online Display 290
12.6.28.1 Scroll Flow Object Class... 290
12.6.28.2 Multi-mode Flow Object Class.. 291
12.6.28.3 Link Flow Object Class... 291
12.6.28.4 Marginalia Flow Object Class 292
Annex A: Further Information.. 293

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 10179 was prepared by Joint Technical Committee ISO/IEC JTC1, *Information technology*.

Introduction

This International Standard defines the Document Style Semantics and Specification Language (DSSSL) used to specify the formatting and transformation of SGML documents. The initial focus of DSSSL is on formatting for both paper and electronic media and on the transformation of SGML documents marked up according to different DTDs. DSSSL may be used with any SGML documents without requiring modifications or constraining the document type definitions.

The main objective of this International Standard is to provide a language for expressing formatting and other document processing specifications in a formal and rigorous manner so that these specifications may be processed by a broad range of formatters, either natively or using a translation mechanism.

The DSSSL style language allows users to specify the types of formatting to be applied to various objects during composition, layout, and pagination. The DSSSL transformation language allows users to specify the transformation of documents from one application of SGML markup into another.

DSSSL is designed for specifications that apply to a class of documents. These specifications are applicable to all possible SGML documents for an SGML application as well as to a particular SGML document.

The DSSSL specification languages are declarative. They are not intended to be complete programming languages, although they contain constructs normally associated with such languages. DSSSL specifications can be unambiguously parsed and interpreted by heterogeneous systems. In addition, DSSSL specifications may be used by existing formatting systems through the use of ‘front-end’ DSSSL processors and translators. DSSSL has no bias toward batch or interactive formatting systems and does not prescribe any pre-defined formatting algorithms.

The standardization of formatting semantics is provided in DSSSL through a set of basic structures known as flow objects and an associated set of formatting characteristics that are applied to those objects. DSSSL provides mechanisms for defining and extending the semantic constructs so that DSSSL application designers can construct DSSSL applications best suited to their application environments.

0.1 Background

The concepts behind DSSSL are associated with the development of generic coding and specifically with SGML, the Standard Generalized Markup Language (ISO 8879).

Historically, electronic manuscripts contained control codes or macro calls that caused the document to be formatted in a particular way (‘specific coding’). In contrast, generic coding, which began in the late 1960s, uses descriptive tags (for example, ‘heading’ rather than ‘Space 3 lines; 14 point Bodoni’). Central to the concept of generic coding is the separation of the information content of documents from the format or appearance of the content. The generic coding concept gained prominence in the early 1970s and came to fruition with the development of SGML.

While SGML provides the language for modeling classes of documents, it does not prescribe any particular model or pre-defined tag set. A set of rules (consisting primarily of a DTD and its supporting documentation) that applies SGML to a class of documents is known as an SGML application.

SGML standardizes the representation of the document structure, leaving it to users to develop their own techniques for interfacing with formatters and other processors, such as general purpose translators. DSSSL is designed to support this second class of applications by providing a standardized architecture for formatting and other processing specifications, allowing users to interchange such specifications within a standardized framework.

A DSSSL specification is normally external to the SGML document to which it applies, and thus multiple specifications may be applied to a given SGML document to yield various presentations of the same data.

SGML provides the ability to distinguish between the intrinsic content and structure of a document, on the one hand, and the specifications for processing it on the other. With DSSSL, formatting and other processing specifications may be interchanged in conjunction with SGML documents to provide the standardized specification of document display while preserving the essential distinction between content and format.