Appendix 1

OID: Z39.50 Object Identifiers

(Normative)

OID.1 Object Identifier Assigned to This Standard

ANSI has assigned the following object identifier to this standard, ANSI-standard-Z39.50:

{iso (1) member-body (2) US (840) ANSI-standard-Z39.50 (10003)}

Note: This OID was originally assigned to Z39.50-1992; it applies also to Z39.50-1995.

OID.2 Object Classes Assigned by This Standard

This standard assigns the following values for object classes, at the level immediately subordinate to ANSI-standard-Z39.50:

1 = application context definitions
2 = abstract syntax definition for APDUs
3 = attribute set definitions
4 = diagnostic definitions
5 = record syntax definitions
6 = transfer syntax definitions
7 = resource report format definitions
8 = access control format definitions
9 = extended services definitions
10 = user information format definitions
11 = element specification format definitions
12 = variant set definitions
13 = database schema definitions
14 = tag set definitions

The following ASN.1 module establishes shorthand notation for the Z39.50 object identifier, and for the object classes. The notation is used in appendices that follow.

```
ANSI-Z39-50-ObjectIdentifier  DEFINITIONS ::=
BEGIN
Z39-50  OBJECT IDENTIFIER ::=
{ iso (1) member-body (2) US (840) ANSI-standard-Z39.50 (10003)}

Z39-50-context          OBJECT IDENTIFIER ::= {Z39-50 1}    -- See Appendix CTX.
Z39-50-APDU             OBJECT IDENTIFIER ::= {Z39-50 2}    -- See OID.3.1.
Z39-50-attributeSet     OBJECT IDENTIFIER ::= {Z39-50 3}    -- See Appendix ATR.
Z39-50-diagnostic       OBJECT IDENTIFIER ::= {Z39-50 4}    -- See Appendix ERR.
Z39-50-recordSyntax     OBJECT IDENTIFIER ::= {Z39-50 5}    -- See Appendix REC.
Z39-50-transferSyntax   OBJECT IDENTIFIER ::= {Z39-50 6}    -- See note below.
Z39-50-resourceReport   OBJECT IDENTIFIER ::= {Z39-50 7}    -- See Appendix RSC.
Z39-50-accessControl    OBJECT IDENTIFIER ::= {Z39-50 8}    -- See Appendix ACC.
Z39-50-extendedService  OBJECT IDENTIFIER ::= {Z39-50 9}    -- See Appendix EXT.
Z39-50-userInfoFormat   OBJECT IDENTIFIER ::= {Z39-50 10}   -- See Appendix USR.
Z39-50-elementSpec      OBJECT IDENTIFIER ::= {Z39-50 11}   -- See Appendix ESP.
Z39-50-variantSet       OBJECT IDENTIFIER ::= {Z39-50 12}   -- See Appendix VAR.
Z39-50-schema           OBJECT IDENTIFIER ::= {Z39-50 13}   -- See Appendix TAG.
Z39-50-tagSet           OBJECT IDENTIFIER ::= {Z39-50 14}   -- See Appendix TAG.
END
```

No object identifier is assigned by this standard for any transfer syntax. For the purpose of presentation context negotiation for an abstract syntax (including the abstract syntax for the APDUs, defined in 4.1), the abstract syntax is paired with a transfer syntax. This pairing is represented by a pair of object identifiers, one for the abstract-syntax (e.g. Z39.50-APDU) and one for the encoding rules. For abstract syntaxes described using ASN.1 (e.g. Z39.50-APDU), a set of basic encoding rules are specified by "ASN.1 Basic Encoding Rules," ISO 8825, identified by the following object identifier:

{ joint-iso-ccitt asn1 (1) basic-encoding (1) }

OID.3 Object Identifiers Assigned by This Standard

All object identifiers assigned by this standard (with the exception of the OID for Z39.50 APDUs, assigned in OID.3.1) are explicitly assigned in the appendices that follow.

OID.3.1 Object Identifiers for Z39.50 APDUs

This standard assigns the following object identifier for the ASN.1 definition of APDUs in 4.1.

Z39-50-APDU **{Z39-50-APDU 1}**

Note: the same OID is used for APDUs both for Z39.50-1992 and Z39.50-1995, because of the interworking capability between the two versions.

OID.4 Object Identifiers Used by This Standard

Z39.50 object identifiers are either public or locally defined. Public Z39.50 object identifiers are officially registered, by this standard or by the Z39.50 maintenance agency (see OID.5). Locally defined Z39.50 object identifiers are registered by a registered Z39.50 implementor (see OID.6 and OID.7).

OID.5 Object Identifiers Assigned by the Z39.50 Maintenance Agency

Additional object identifiers (official Z39.50 object identifiers not registered by this standard) may be assigned by the Z39.50 Maintenance Agency (see note), of the form: {Z39-50 n m} where {z39-50 n} is an object class defined in OID.2, or is an additional object class defined by the maintenance agency.
Note: At the time of approval of this standard, the Z39.50 Maintenance Agency is the Library of Congress.

OID.6 Locally Registered Objects

Locally registered objects are of the form:

{Z39-50 n 1000 p m}

where {z39-50 n} is as described in OID.5, and 'p' is the OID index of a registered Z39.50 Implementor (contact the Z39.50 Maintenance Agency for procedures for registration of an implementor). A locally registered object may be published or private. Local, published objects are those whose definitions are coordinated with and published by the Z39.50 Maintenance Agency. Local, private objects are those whose definitions are not published by the Z39.50 Maintenance Agency.

OID.7 Experimental Objects

Experimental objects are of the form:

{Z39-50 n 2000 p m}

where {z39-50 n} is as described in OID.5, and 'p' is the OID index of a registered Z39.50 Implementor.

Appendix 2:

CTX: Application Context basic-Z39.50-ac

(Normative)

This standard defines and registers the application context basic-Z39.50-ac. The object identifier for application context basic-Z39.50-ac is:

{ Z39-50-context 1 }

Definition of application context basic-Z39.50-ac

ANSI-standard-Z39.50 application context basic-Z39.50-ac supports an application-entity that contains only the following two application service elements (ASEs):

a) the association control service element (ACSE, ISO 8650), and
b) the Z39.50 service element.

Z39.50 and ACSE are used according to the procedures in section 4.2.1.

The presentation services required are those contained in the presentation kernel functional unit and the session duplex functional unit. All Information Retrieval protocol data units will be mapped onto the P-Data service.

In the event of protocol errors, the system detecting the error shall abort the association.

Only the origin may invoke the A-RELEASE service (to initiate orderly release of an A-association).

Appendix 3

ATR: Attribute Sets

(Normative)

This standard registers the attribute sets listed below, and assigns the following object identifiers:

Bib-1	{Z39-50-attributeSet 1}	(See ATR.1)
Exp-1	{Z39-50-attributeSet 2}	(See ATR.2)
Ext-1	{Z39-50-attributeSet 3}	(See ATR.3)
CCL-1	{Z39-50-attributeSet 4}	
GILS	{Z39-50-attributeSet 5}	
STAS	{Z39-50-attributeSet 6}	

Each attribute set defines a set of types and for each type a set of values. An attribute list (see AttributeList in the ASN.1 for APDUs, 4.1), constructed from an attribute set definition, is a list of attribute pairs. An attribute pair (AttributeElement in the ASN.1 for APDUs) consists of an attribute type and a value list (attributeValue within AttributeElement), where each value in the list is defined for that type.

When version 2 is in force, each value list is a single value and is an integer. When version 3 is in force, attributeValue (within AttributeElement) may select 'complex', allowing the value list to include multiple values (each may be integer or string) and also to specify a 'semanticAction', indicating how the target is to treat the multiple attributes.

When an attribute list contains any attribute pair where attributeValue selects 'complex', there must not be any attribute type within the attribute list for which there is more than a single attribute pair.

ATR.1 Attribute Set bib-1

This section defines the attribute set bib-1.

ATR.1.1 Bib-1 Types and Values

This section lists the attribute types and values for attribute-set bib-1 (see Tables A-3-1 through A-3-6).

Attribute Type	Value	Attribute Type	Value
Use	1	Structure	4
Relation	2	Truncation	5
Position	3	Completeness	6

Table A-3-1: Bib-1 Use Attributes

Use	Value	Use	Value	Use	Value
Personal name	1	Subject heading	21	Title other variant	41
Corporate name	2	Subject Rameau	22	Title former	42
Conference name	3	BDI index subject	23	Title abbreviated	43
Title	4	INSPEC subject	24	Title expanded	44
Title series	5	MESH subject	25	Subject precis	45
Title uniform	6	PA subject	26	Subject rswk	46
ISBN	7	LC subject heading	27	Subject subdivision	47
ISSN	8	RVM subject heading	28	No. nat'l biblio.	48
LC card number	9	Local subject index	29	No. legal deposit	49
BNB card no.	10	Date	30	No. govt pub.	50
BGF number	11	Date of publication	31	No. music publisher	51
Local number	12	Date of acquisition	32	Number db	52
Dewey classification	13	Title key	33	Number local call	53
UDC classification	14	Title collective	34	Code--language	54
Bliss classification	15	Title parallel	35	Code--geographic area	55
LC call number	16	Title cover	36	Code--institution	56
NLM call number	17	Title added title page	37	Name and title	57
NAL call number	18	Title caption	38	Name geographic	58
MOS call number	19	Title running	39	Place publication	59
Local classification	20	Title spine	40	CODEN	60

Table A-3-1: Bib-1 Use Attributes (continued)

Use	Value	Use	Value	Use	Value
Microform generation	61	Body of text	1010	Indexed-by	1023
Abstract	62	Date/time added to db	1011	Map-scale	1024
Note	63	Date/time last modified	1012	Music-key	1025
Author-title	1000	Authority/format id	1013	Related-periodical	1026
Record type	1001	Concept-text	1014	Report-number	1027
Name	1002	Concept-reference	1015	Stock-number	1028
Author	1003	Any	1016	Thematic-number	1030
Author-name personal	1004	Server-choice	1017	Material-type	1031
Author-name corporate	1005	Publisher	1018	Doc-id	1032
Author-name conference	1006	Record-source	1019	Host-item	1033
Identifier--standard	1007	Editor	1020	Content-type	1034
Subject--LC children's	1008	Bib-level	1021	Anywhere	1035
Subject name -- personal	1009	Geographic-class	1022	Author-Title-Subject	1036

Table A-3-2: Bib-1 Relation Attributes

Relation	Value	Relation	Value	Relation	Value
less than	1	greater or equal	4	phonetic	100
less than or equal	2	greater than	5	stem	101
equal	3	not equal	6	relevance	102
				AlwaysMatches	103

Table A-3-3: Bib-1 Position Attributes

Position	Value	Position	Value	Position	Value
first in field	1	first in subfield	2	any position in field	3

Table A-3-4: Structure Attribute

Structure	Value	Structure	Value	Structure	Value
phrase	1	word list	6	urx	104
word	2	date (un-normalized)	100	free-form-text	105
key	3	name (normalized)	101	document-text	106
year	4	name (un-normalized)	102	local number	107
date (normalized)	5	structure	103	string	108
				numeric string	109

Table A-3-5: Bib-1 Truncation Attributes

Truncation	Value	Truncation	Value	Truncation	Value
right Truncation	1	do not truncate	100	regExpr-1	102
left truncation	2	process # in search term	101	regExpr-2	103
left and right	3				

Table A-3-6: Bib-1 Completeness Attributes

Completeness	Value	Completeness	Value	Completeness	Value
incomplete subfield	1	complete subfield	2	complete field	3

ATR.1.2 Bib-1 Attribute Combinations

If a target does not support a given attribute list, it should fail the search and supply an appropriate diagnostic.

A given attribute type may appear zero times, one time, or more than one time, in an attribute list.

- If an attribute type does not occur in an attribute list, then (in the absence of any prior understanding, either outside of the standard or via the Explain facility) the origin should not expect any particular default target behavior.
- If an attribute type occurs exactly once in an attribute list, then the attribute value specifies the preferred target behavior with respect to that attribute type.
- It is recommended that an attribute type not occur more than once in an attribute list, unless an associated "semantic action" is included (i.e. attributeValue selects 'complex').

When attributeValue selects 'complex', 'semanticAction' may be included. SemanticAction is a sequence of integers; for bib-1, it is either a single integer or a sequence of two integers.

For the first integer in the sequence, values are:

1 May not substitute another attribute. If none in the list is supported, fail the search.
2 May substitute another attribute, but only if none in the list is supported.
3 May substitute another attribute at target discretion (even if one or more in the list is supported).

The second integer in the sequence is to be supplied if and only if there are multiple attributes in the list. Values are:

1 Select the first supported attribute in the list.
2 Select the best attribute in the list.

ATR.2 Attribute Set exp-1

This section defines the attribute-set exp-1, for use with an Explain database. The attribute set exp-1 defines a single attribute type, 'Use'. In addition, this attribute set definition *imports* non-Use bib-1 attributes, i.e. those of type Relation, Position, Structure, Truncation, and Completeness (see tables A-3-2 through A-3-6). The types and values defined within the bib-1 attribute set for these attributes may be used within the exp-1 attribute set, using the object identifier for this attribute set. It is recommended that a target supporting the Explain facility support the Relation attribute 'equal', Position attribute 'any' position in field', and Structure attribute 'key'.

Note: If the target supports searching based on date ranges (e.g. to limit a search to records created before or after a particular date or between two dates), the target should also support one or more of the following relation attributes: 'less than', 'less than or equal', 'greater than', and 'greater or equal'.

Table A-3-7: Exp-1 Use Attributes

Use	Value	Use	Value	Use	Value
ExplainCategory	1	DateChanged	10	Availability	19
HumanStringLanguage	2	DateExpires	11	Proprietary	20
DatabaseName	3	ElementSetName	12	UserFee	21
TargetName	4	ProcessingContext	13	VariantSetOID	22
AttributeSetOID	5	ProcessingName	14	UnitSystem	23
RecordSyntaxOID	6	TermListName	15	Keyword	25
TagSetOID	7	SchemaOID	16	ExplainDatabase	26
ExtendedServiceOID	8	Producer	17	ProcessingOID	27
DateAdded	9	Supplier	18		

Notes:

(1) The search terms for Use attribute ExplainCategory are listed in table A-3-8.

(2) The search term when the Use attribute is HumanStringLanguage is the three-character language code from ANSI/NISO Z39.53-1994 -- Codes for the Representation of Languages for Information Interchange.

(3) The search terms when the Use attribute is ProcessingContext are listed in table A-3-9.

(4) Where the search term is an object identifier (where the name of the Use attribute ends with "OID"): for version 2, it is recommended that the term be a character string representing a sequence of integers (each represented by a character string) separated by periods. For version 3, it is recommended that the term be represented as ASN.1 type OBJECT IDENTIFIER.

(5) Use attribute Keyword is used when searching for DatabaseInfo records (i.e. in combination with an operand where Use is ExplainCategory and term is DatabaseInfo). Its use is to search in the keyword element, for terms that match one of the query terms.

(6) Use attribute ExplainDatabase is used when searching for DatabaseInfo records (i.e. in combination with an operand where Use is ExplainCategory and term is DatabaseInfo). The term should be NULL, for version 3, or otherwise ignored by the target. The Relation attribute either should be omitted or should be AlwaysMatches.

Table A-3-8: Search terms associated with use attribute ExplainCategory

TargetInfo	TermListInfo	SortDetails
DatabaseInfo	extendedServicesInfo	Processing
SchemaInfo	AttributeDetails	CategoryList
TagSetInfo	TermListDetails	VariantSetInfo
RecordSyntaxInfo	ElementSetDetails	UnitInfo
AttributeSetInfo	RetrievalRecordDetails	

Table A-3-9: Search terms associated with use attribute ProcessingContext

Access	Retrieval	RecordHandling
Search	RecordPresentation	

ATR.3 Attribute Set ext-1

This section defines the attribute-set ext-1, for use with an Extended Services database (see Tables A-3-10 and A-3-11). Two types are defined:

Attribute Type	Value
Use	1
Permissions	2

Additional attributes (types and/or values) may be defined within a specific Extended Service definition. The attribute set id to be used to identify those attributes is the ObjectIdentifier that identifies the specific Extended Service.

Table A-3-10: Ext-1 Use Attributes

Use	Value	Use	Value	Use	Value
UserId	1	TaskStatus	4	RetentionTime	6
PackageName	2	PackageType	5	TargetReference	7
CreationDatetime	3				

Table A-3-11: Ext-1 Permission Attributes

Permission	Value	Permission	Value	Permission	Value
Delete	1	ModifyPermissions	3	Invoke	5
Modify	2	Present	4	Any	6

Note: The Permission attribute is for use only when the value of the Use attribute is UserId, in which case the purpose is to search for task packages for which the specified user has the specified permission.

Appendix 4

ERR: Error Diagnostics

(Normative)

This standard defines and registers the diagnostic set bib-1 and the diagnostic format diag-1. The following object identifiers are assigned:

bib-1 {Z39-50-diagnostic 1} (See ERR.1)
diag-1 {Z39-50-diagnostic 2} (See ERR.2)

When version 2 is in force, a diagnostic record must conform to the following format:

```
DefaultDiagFormat ::= SEQUENCE{
        diagnosticSetId     OBJECT IDENTIFIER,
        condition   INTEGER,
        addinfo     VisibleString}
```

The diagnostic record includes an integer corresponding to a condition or error, and an (object) identifier of the diagnostic set definition that lists the condition corresponding to that integer.

When version 3 is in force, a diagnostic record may assume the form above, or alternatively, may be defined as EXTERNAL, identified by an OBJECT IDENTIFIER (which identifies the diagnostic *format*, rather than the diagnostic *set*).

Bib-1 is a diagnostic *set*. It was originally defined and registered in Z39.50-1992. This standard includes an extended definition; the conditions listed below for bib-1 include all those that were listed in the Z39.50-1992 bib-1 definition, as well as several additional diagnostics that have been added. (In particular, several of the conditions described by diag-1 that can be expressed by the above format have been added to bib-1.)

Diag-1 is a diagnostic *format*. It includes several structures for diagnostic information, each tailored to the error information being described. It also includes a single structure through which a diagnostic from a diagnostic set (e.g. bib-1) can be referenced.

Diag-1 allows several diagnostic conditions within a single diagnostic record, to describe multiple errors pertaining to the same record or operation. In particular, diagnostics from different diagnostic sets may be included within the same diag-1 diagnostic record.

ERR.1 Diagnostic Set Bib-1

Table A-4-1 below is for use when DiagnosticSetId (within DefaultDiagFormat) equals the object identifier for diagnostic set bib-1, in which case, Condition takes values from the "Code" column below.

This table may also be used by diagnostic format diag-1 when defaultDiagRec is selected for "diagnostic," and DiagnosticSetId equals the object identifier for this diagnostic set. In that case, the values of "code" and "addinfo" are taken from this table.

AddInfo is ASN.1 type VisibleString. However, for several of the diagnostics below, AddInfo is used to express the value of a parameter that has an ASN.1 type other than VisibleString. Where Addinfo is used to express a numeric value, it should be a character string representation of that value. Where Addinfo is used to express an object identifier, it should take the form of a sequence of integers (each represented by a character string) separated by periods.

Table A-4-1: Diagnostic Conditions

code	Meaning	Addinfo
1	permanent system error	(unspecified)
2	temporary system error	(unspecified)
3	unsupported search	(unspecified)
4	Terms only exclusion (stop) words	(unspecified)
5	Too many argument words	(unspecified)
6	Too many boolean operators	(unspecified)
7	Too many truncated words	(unspecified)
8	Too many incomplete subfields	(unspecified)
9	Truncated words too short	(unspecified)
10	Invalid format for record number (search term)	(unspecified)
11	Too many characters in search statement	(unspecified)
12	Too many records retrieved	(unspecified)

Table A-4-1: Diagnostic Conditions (continued)

code	Meaning	Addinfo
13	Present request out-of-range	(unspecified)
14	System error in presenting records	(unspecified)
15	Record not authorized to be sent intersystem	(unspecified)
16	Record exceeds Preferred-message-size	(unspecified)
17	Record exceeds Exceptional-record-size	(unspecified)
18	Result set not supported as a search term	(unspecified)
19	Only *single* result set as search term supported	(unspecified)
20	Only *AND*ing of a *single* result set as search term	(unspecified)
21	Result set exists and replace indicator off	(unspecified)
22	Result set naming not supported	(unspecified)
23	Specified combination of databases not supported	(unspecified)
24	Element set names not supported	(unspecified)
25	Specified element set name not valid for specified database	(unspecified)
26	Only generic form of element set name supported	(unspecified)
27	Result set no longer exists - unilaterally deleted by target	(unspecified)
28	Result set is in use	(unspecified)
29	One of the specified databases is locked	(unspecified)
30	Specified result set does not exist	(unspecified)
31	Resources exhausted - no results available	(unspecified)
32	Resources exhausted - unpredictable partial results available	(unspecified)
33	Resources exhausted - valid subset of results available	(unspecified)
100	(unspecified) error	(unspecified)
101	Access-control failure	(unspecified)
102	Challenge required, could not be issued - operation terminated	(unspecified)
103	Challenge required, could not be issued - record not included	(unspecified)
104	Challenge failed - record not included	(unspecified)
105	Terminated at origin request	(unspecified)
106	No abstract syntaxes agreed to for this record	(unspecified)
107	Query type not supported	(unspecified)
108	Malformed query	(unspecified)
109	Database unavailable	database name
110	Operator unsupported	operator
111	Too many databases specified	maximum
112	Too many result sets created	maximum
113	Unsupported attribute type	type
114	Unsupported Use attribute	value
115	Unsupported *term* value for Use attribute	term
116	Use attribute required but not supplied	(unspecified)
117	Unsupported Relation attribute	value
118	Unsupported Structure attribute	value
119	Unsupported Position attribute	value
120	Unsupported Truncation attribute	value
121	Unsupported Attribute Set	oid
122	Unsupported Completeness attribute	value
123	Unsupported attribute combination	(unspecified)
124	Unsupported coded value for term	value
125	Malformed search term	(unspecified)
126	Illegal term value for attribute	term
127	Unparsable format for un-normalized value	value
128	Illegal result set name	name
129	Proximity search of sets not supported	(unspecified)

Table A-4-1: Diagnostic Conditions (continued)

code	Meaning	Addinfo
130	Illegal result set in proximity search	result set name
131	Unsupported proximity relation	value
132	Unsupported proximity unit code	value
201	Proximity not supported with this attribute combination	attribute list
202	Unsupported distance for proximity	distance
203	Ordered flag not supported for proximity	(unspecified)
205	Only zero step size supported for Scan	(unspecified)
206	Specified step size not supported for Scan	step size
207	Cannot sort according to sequence	sequence
208	No result set name supplied on Sort	(unspecified)
209	Generic sort not supported (database-specific sort only supported)	(unspecified)
210	Database specific sort not supported	(unspecified)
211	Too many sort keys	number
212	Duplicate sort keys	key
213	Unsupported missing data action	value
214	Illegal sort relation	relation
215	Illegal case value	value
216	Illegal missing data action	value
217	Segmentation: Cannot guarantee records will fit in specified segments	(unspecified)
218	ES: Package name already in use	name
219	ES: no such package, on modify/delete	name
220	ES: quota exceeded	(unspecified)
221	ES: extended service type not supported	type
222	ES: permission denied on ES - id not authorized	(unspecified)
223	ES: permission denied on ES - cannot modify or delete	(unspecified)
224	ES: immediate execution failed	(unspecified)
225	ES: immediate execution not supported for this service	(unspecified)
226	ES: immediate execution not supported for these parameters	(unspecified)
227	No data available in requested record syntax	(unspecified)
228	Scan: malformed scan	(unspecified)
229	Term type not supported	type
230	Sort: too many input results	max
231	Sort: incompatible record formats	(unspecified)
232	Scan: term list not supported	alternative term list
233	Scan: unsupported value of position-in-response	value
234	Too many index terms processed	number of terms
235	Database does not exist	database name
236	Access to specified database denied	database name
237	Sort: illegal sort	(unspecified)
238	Record not available in requested syntax	alternative suggested syntax(es)
239	Record syntax not supported	syntax
240	Scan: Resources exhausted looking for satisfying terms	(unspecified)
241	Scan: Beginning or end of term list	(unspecified)
242	Segmentation: max-segment-size too small to segment record	smallest acceptable size
243	Present: additional-ranges parameter not supported	(unspecified)
244	Present: comp-spec parameter not supported	(unspecified)
245	Type-1 query: restriction ('resultAttr') operand not supported	(unspecified)
246	Type-1 query: 'complex' attributeValue not supported	(unspecified)
247	Type-1 query: 'attributeSet' as part of AttributeElement not supported	(unspecified)

ERR.2 Diagnostic Format Diag-1

This section defines the diagnostic format diag-1.

```
DiagnosticFormatDiag1
{Z39-50-diagnosticFormat diag-1 (2)} DEFINITIONS ::=
BEGIN
IMPORTS Term, Specification, AttributeList, SortElement, DatabaseName,
DefaultDiagFormat, InternationalString FROM Z39-50-APDU-1995;


DiagnosticFormat ::=  SEQUENCE OF SEQUENCE{
   diagnostic   [1] CHOICE{
                      defaultDiagRec        [1]    IMPLICIT DefaultDiagFormat,
                      explicitDiagnostic    [2]    DiagFormat} OPTIONAL,
   message  [2] IMPLICIT InternationalString OPTIONAL}


DiagFormat ::= CHOICE{

tooMany     [1000]    IMPLICIT SEQUENCE{
                         tooManyWhat  [1]    IMPLICIT INTEGER{
                                               argumentWords           (1),
                                               truncatedWords          (2),
                                               booleanOperators        (3),
                                               incompleteSubfields     (4),
                                               characters              (5),
                                               recordsRetrieved        (6),
                                               dataBasesSpecified      (7),
                                               resultSetsCreated       (8),
                                               indexTermsProcessed     (9)},
                         max          [2]    IMPLICIT INTEGER OPTIONAL},


badSpec     [1001]    IMPLICIT SEQUENCE{   -- element set name or specification
                         spec         [1] IMPLICIT Specification, -- esn or element spec not supported
                         db           [2] IMPLICIT DatabaseName OPTIONAL,
                                         -- if db specified, above spec not supported for db; otherwise,
                                         -- spec not supported period.
                         goodOnes     [3] IMPLICIT SEQUENCE OF Specification OPTIONAL
                                         -- target supplies ones that are supported
                                         },


dbUnavail   [1002]    IMPLICIT SEQUENCE{ -- database unavailable
                         db     [1] IMPLICIT DatabaseName,
                         why    [2] IMPLICIT SEQUENCE{
                                   reasonCode [1] IMPLICIT INTEGER{
                                                     doesNotExist       (0),
                                                     existsButUnavail   (1),
                                                     locked             (2),
                                                     accessDenied       (3)} OPTIONAL,
                                   message    [2] IMPLICIT InternationalString OPTIONAL}},
```

```
unSupOp     [1003]   IMPLICIT INTEGER{  -- unsupported operator
                       and       (0),
                       or        (1),
                       and-not   (2),
                       prox      (3)},


attribute   [1004]   IMPLICIT SEQUENCE{
                          -- Applies for unsupported attribute set, attribute type,
                          -- attribute value, or term (for a given attribute type or value).

                       id        [1]    IMPLICIT OBJECT IDENTIFIER,
                          -- if only "id" occurs, then attribute set is not supported
                       type      [2]    IMPLICIT INTEGER OPTIONAL,
                          -- must occur if value occurs.
                       value     [3]    IMPLICIT INTEGER OPTIONAL,
                          -- if omitted, and Type occurs, then Type is what is unsupported
                       term      [4]    Term OPTIONAL
                          -- If occurs, term is illegal or not supported, for attribute value,
                          -- if value occurs; otherwise, for type.
                                           },


attCombo    [1005]   IMPLICIT SEQUENCE{ -- attribute combination not supported
                     unsupportedCombination     [1] IMPLICIT AttributeList,
                     recommendedAlternatives    [2] IMPLICIT SEQUENCE OF AttributeList OPTIONAL},


term        [1006]   IMPLICIT SEQUENCE{
                       problem      [1] IMPLICIT INTEGER{
                                          codedValue (1),
                                          unparsable (2),
                                          tooShort   (3),
                                          type       (4)} OPTIONAL,
                       term         [2] Term},


proximity   [1007] CHOICE{             -- proximity diagnostics:
                     resultSets   [1]   IMPLICIT NULL,                -- proximity between sets not supported
                     badSet       [2]   IMPLICIT InternationalString, -- bad result set specified
                     relation     [3]   IMPLICIT INTEGER,             -- 1 to 6 ; relation not supported
                     unit         [4]   IMPLICIT INTEGER,             -- unsupported unit code
                     distance     [5]   IMPLICIT INTEGER,             -- unsupported distance
                     attributes   [6]   AttributeList,                --   proximity not supported with specified
                                                                      -- attribute combination
                     ordered      [7]   IMPLICIT NULL,                -- ordered flag not supported
                     exclusion    [8]   IMPLICIT NULL                 -- exclusion flag not supported
                            },
```

```
scan        [1008] CHOICE{          -- scan diagnostics:
                nonZeroStepSize  [0] IMPLICIT NULL,     -- only zero step size supported
                specifiedStepSize [1] IMPLICIT NULL,    -- specified step size not supported
                termList1        [3] IMPLICIT NULL,     -- term list not supported (no alternative supplied)
                termList2        [4] IMPLICIT SEQUENCE OF AttributeList,
                                                        -- term list not supported (alternatives supplied)
                posInResponse    [5] IMPLICIT INTEGER{   --value of positionInResponse not supported
                                    mustBeOne           (1),
                                    mustBePositive      (2),
                                    mustBeNonNegative   (3),
                                    other               (4)},
                resources        [6] IMPLICIT NULL,     -- resources exhausted looking for satisfying terms
                endOfList        [7] IMPLICIT NULL      -- beginning or end of term list
                  },


sort          [1009] CHOICE{
                sequence      [0]   IMPLICIT NULL,             -- cannot sort according to sequence
                noRsName      [1]   IMPLICIT NULL,             -- no result set name supplied
                tooMany       [2]   IMPLICIT INTEGER,          --  Too many input result sets,
                                                               --  maximum supplied.
                incompatible  [3]   IMPLICIT NULL,             -- records with different formats
                                                               -- not compatible for sorting
                generic       [4]   IMPLICIT NULL,             -- generic sort not supported
                                                               -- (db specific only)
                dbSpecific    [5]   IMPLICIT NULL,             -- db specific sort not supported
                sortElement   [6]   SortElement,
                key           [7]   IMPLICIT INTEGER{
                                       tooMany     (1),      -- too many sort keys
                                       duplicate   (2)},     -- duplicate sort keys
                action        [8]   IMPLICIT NULL,     -- unsupported missing data action
                illegal       [9]   IMPLICIT INTEGER{
                                       relation  (1),   -- illegal sort relation
                                       case      (2),   -- illegal case value
                                       action    (3),   -- illegal missing data action
                                       sort      (4)},  -- illegal sort
             inputTooLarge    [10]  IMPLICIT SEQUENCE OF InternationalString,
                                                  --   one or more of the input result sets too large to sort
           aggregateTooLarge  [11]  IMPLICIT NULL --  aggregate result set too large
                                  },

segmentation   [1010]   CHOICE{
                   segmentCount  [0]       IMPLICIT NULL,
                                              -- Cannot guarantee record will fit within max segments. Target
                                              -- suggests that origin try again to retrieve record, without
                                              -- including max-segment-count.
                   segmentSize   [1]       IMPLICIT INTEGER
                                           -- record cannot be segmented into fragments such that the largest
                                           -- will fit within max segment size specified. Target supplies (in
                                           -- bytes) the smallest acceptable value of Max-segment-size to
                                           -- retrieve the record.
                                  },
```

```
extServices  [1011] CHOICE{
                    req            [1] IMPLICIT INTEGER{    -- bad request
                                   nameInUse      (1),    -- package name already in use
                                   noSuchName     (2),    -- no such package, on modify/delete
                                   quota          (3),    -- quota exceeded
                                   type           (4)},   -- extended service type not supported
                    permission  [2] IMPLICIT INTEGER{ -- permission denied on ES, because:
                                   id             (1),        --   id not authorized, or
                                   modifyDelete   (2)},       -- cannot modify or delete
                    immediate   [3] IMPLICIT INTEGER{ -- immediate execution:
                                   failed         (1),        -- failed,
                                   service        (2),        -- not supported for this service, or
                                   parameters     (3)         -- for these parameters.
                                        }},


accessCtrl       [1012] CHOICE{
                    noUser       [1] IMPLICIT NULL,         -- no user to display challenge to
                    refused      [2] IMPLICIT NULL,         -- access control information refused by user
                    simple       [3] IMPLICIT NULL,         -- only simple form supported (target used
                                                            -- externally defined)
                    oid          [4] IMPLICIT SEQUENCE OF OBJECT IDENTIFIER,
                                                            -- oid not supported (origin supplies alternative
                                                            -- suggested oids)
                    alternative  [5] IMPLICIT SEQUENCE OF OBJECT IDENTIFIER,
                                                            -- origin insists that target use an alternative
                                                            -- challenge for this data (e.g. stronger
                                                            -- authentication or stronger Access control). The
                                                            -- origin supplies suggested alternative oids.
                    pwdInv       [6] IMPLICIT NULL,         -- password invalid
                    pwdExp       [7] IMPLICIT NULL          -- password expired
                         },


recordSyntax     [1013] IMPLICIT SEQUENCE{ -- record cannot be transferred in requested syntax
                  unsupportedSyntax        [1] IMPLICIT OBJECT IDENTIFIER,
                  suggestedAlternatives    [2] IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL}
}

END
```

Appendix 5

REC: Record Syntaxes

(Normative)

This standard registers the following object identifiers for record syntaxes:

Object identifiers assigned for bibliographic syntaxes, not *described via ASN.1:*

Unimarc	{Z39-50-recordSyntax	1	}
Intermarc	{Z39-50-recordSyntax	2	}
CCF	{Z39-50-recordSyntax	3	}
USmarc	{Z39-50-recordSyntax	10	}
UKmarc	{Z39-50-recordSyntax	11	}
Normarc	{Z39-50-recordSyntax	12	}
Librismarc	{Z39-50-recordSyntax	13	}
Danmarc	{Z39-50-recordSyntax	14	}
Finmarc	{Z39-50-recordSyntax	15	}
MAB	{Z39-50-recordSyntax	16	}
Canmarc	{Z39-50-recordSyntax	17	}
SBN	{Z39-50-recordSyntax	18	}
Picamarc	{Z39-50-recordSyntax	19	}
Ausmarc	{Z39-50-recordSyntax	20	}
Ibermarc	{Z39-50-recordSyntax	21	}

Note: The following transfer syntax (see ISO 2709) may be used in conjunction with the above bibliographic definitions:

ISO2709 {iso standard 2709 transfer-syntax (1) character-encoding (1)}

When presentation context negotiation is used, the above syntaxes may be paired with the transfer syntax identified by the object identifier ISO2709 for the transfer-syntax for bibliographic records defined in ISO 2709. When presentation context negotiation is not used, the above record syntaxes are assumed to be paired with ISO2709.

Object identifiers assigned for syntaxes which are *described via ASN.1:*

Explain	{Z39-50-recordSyntax 100}	(See REC.1)
SUTRS	{Z39-50-recordSyntax 101}	(See REC.2)
OPAC	{Z39-50-recordSyntax 102}	(See REC.3)
Summary	{Z39-50-recordSyntax 103}	(See REC.4)
GRS-1	{Z39-50-recordSyntax 105}	(See REC.5)
Extended Services	{Z39-50-recordSyntax 106}	(See REC.6)

Note: The following transfer syntax (see ISO 8825) may be used in conjunction with these definitions:

ISO8825 ::= OBJECT IDENTIFIER
{joint-iso-ccitt (2) basic-encoding (1)}

When presentation context negotiation is used, these syntaxes may be paired with the transfer syntax identified by the object identifier ISO8825 for the transfer-syntax defined in ISO 8825. When presentation context negotiation is not used, the above record syntaxes are assumed to be paired with ISO8825.

REC.1 Explain Record Syntax

```
RecordSyntax-explain
{Z39-50-recordSyntax explain (100)} DEFINITIONS ::=

BEGIN
IMPORTS AttributeSetId, Term, OtherInformation, DatabaseName, ElementSetName, IntUnit, Unit,
 StringOrNumeric, Specification, InternationalString, AttributeList, AttributeElement FROM Z39-50-APDU-1995;

Explain-Record ::= CHOICE{
    -- Each of these may be used as search term  when Use attribute is 'explain-category'.
  targetInfo              [0]     IMPLICIT TargetInfo,
  databaseInfo            [1]     IMPLICIT DatabaseInfo,
  schemaInfo              [2]     IMPLICIT SchemaInfo,
  tagSetInfo              [3]     IMPLICIT TagSetInfo,
  recordSyntaxInfo        [4]     IMPLICIT RecordSyntaxInfo,
  attributeSetInfo        [5]     IMPLICIT AttributeSetInfo,
```

```
  termListInfo              [6]      IMPLICIT TermListInfo,
  extendedServicesInfo      [7]      IMPLICIT ExtendedServicesInfo,
  attributeDetails          [8]      IMPLICIT AttributeDetails,
  termListDetails           [9]      IMPLICIT TermListDetails,
  elementSetDetails         [10]     IMPLICIT ElementSetDetails,
  retrievalRecordDetails    [11]     IMPLICIT RetrievalRecordDetails,
  sortDetails               [12]     IMPLICIT SortDetails,
  processing                [13]     IMPLICIT ProcessingInformation,
  variants                  [14]     IMPLICIT VariantSetInfo,
  units                     [15]     IMPLICIT UnitInfo,
  categoryList              [100]    IMPLICIT CategoryList}

-- Element set name 'B' (brief)  retrieves:
--   -  'commonInfo' (except for otherInfo within commonInfo)
--   -  key elements
--   -  other elements designated as 'non-key brief elements'
-- Esn 'description' retrieves brief elements as well as 'description', and specific additional descriptive
-- elements if designated.
-- Element set name 'F' (full) retrieves all of the above, as well as those designated as "non-brief elements". Some
-- elements designated as OPTIONAL may be mandatory in full records, and are so identified. (Note that all elements
-- that are not part of the brief element set must be designated as OPTIONAL in the ASN.1, otherwise it would be
-- illegal to omit them.)
-- Other esns are defined (below) as needed.

-- - - - - - - - - - - - -  Info Records
  -- Info records are mainly for software consumption
  -- They describe individual entities within the target system:
  --   The target itself
  --   Individual databases
  --   Schemas
  --   Tag sets
  --   Record syntaxes
  --   Attribute sets
  --   Term lists
  --   Extended services
 --       The information about each Schema, Tag Set, Record Syntax and Attribute Set should
  -- match the universal definitions of these items. The only exception is that a target may omit any
  -- items it doesn't support, for example the description of the bib-1 attribute set may omit attributes
  -- that the target does not support under any circumstances.
  --     Databases that may be searched together can be listed in the dbCominations element of the TargetInfo record.

TargetInfo ::= SEQUENCE {
  commonInfo                [0]      IMPLICIT CommonInfo OPTIONAL,
     -- Key elements follow:
  name                      [1]      IMPLICIT InternationalString,
     -- Non-key brief elements follow:
  recent-news               [2]      IMPLICIT HumanString OPTIONAL,
  icon                      [3]      IMPLICIT IconObject OPTIONAL,
  namedResultSets           [4]      IMPLICIT BOOLEAN,
  multipleDBsearch          [5]      IMPLICIT BOOLEAN,
  maxResultSets             [6]      IMPLICIT INTEGER OPTIONAL,
  maxResultSize             [7]      IMPLICIT INTEGER OPTIONAL,
  maxTerms                  [8]      IMPLICIT INTEGER OPTIONAL,
```

```
  timeoutInterval          [9]      IMPLICIT IntUnit OPTIONAL,
  welcomeMessage           [10]     IMPLICIT HumanString OPTIONAL,
    -- non-brief elements follow:
    -- 'description' esn retrieves the following two (as well as brief):
  contactInfo              [11]     IMPLICIT ContactInfo OPTIONAL,
  description              [12]     IMPLICIT HumanString OPTIONAL,
  nicknames                [13]     IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
  usage-restrictions       [14]     IMPLICIT HumanString OPTIONAL,
  paymentAddr              [15]     IMPLICIT HumanString OPTIONAL,
  hours                    [16]     IMPLICIT HumanString OPTIONAL,
  dbCombinations           [17]     IMPLICIT SEQUENCE OF DatabaseList OPTIONAL,
  addresses                [18]     IMPLICIT SEQUENCE OF NetworkAddress OPTIONAL,
  languages                [101]    IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
                                        -- Languages supported for message strings.  Each is a three-character
                                        -- language code from Z39.53-1994.
-- characterSets           [102]  this tag reserved for "character sets supported for name and message strings".
  -- commonAccessInfo elements list objects the target supports. All objects listed in
  -- AccessInfo for any individual database should also be listed here.
  commonAccessInfo         [19]     IMPLICIT AccessInfo OPTIONAL}


DatabaseInfo ::= SEQUENCE {
              -- A target may provide "virtual databases" that are combinations of individual database. These
              -- databases are indicated by the presence of subDbs in the combination database's DatabaseDescription.
  commonInfo                  [0]      IMPLICIT CommonInfo OPTIONAL,
   -- Key elements follow:
  name                        [1]      IMPLICIT DatabaseName,
   -- Non-key brief elements follow:
  explainDatabase             [2]      IMPLICIT NULL OPTIONAL,
                                           -- If present, this database is the Explain database, or an Explain database
                                           -- for a different server, possibly on a different host. The means by which
                                           -- that server may be accessed is not addressed by this standard. One
                                           -- suggested possibility is an implementor agreement whereby the
                                           -- database name is a url which may be used to connect to the server.
  nicknames                   [3]      IMPLICIT SEQUENCE OF DatabaseName OPTIONAL,
  icon                        [4]      IMPLICIT IconObject OPTIONAL,
  user-fee                    [5]      IMPLICIT BOOLEAN,
  available                   [6]      IMPLICIT BOOLEAN,
  titleString                 [7]      IMPLICIT HumanString OPTIONAL,
   -- Non-brief elements follow:
  keywords                    [8]      IMPLICIT SEQUENCE OF HumanString OPTIONAL,
  description                 [9]      IMPLICIT HumanString OPTIONAL,
  associatedDbs               [10]     IMPLICIT DatabaseList OPTIONAL,
                                           -- databases that may be searched in combination with this one

  subDbs                      [11]     IMPLICIT DatabaseList OPTIONAL,
                                           -- When present, this database is a composite representing the combined
                                           -- databases 'subDbs'. The individual subDbs are also available.
  disclaimers                 [12]     IMPLICIT HumanString OPTIONAL,
  news                        [13]     IMPLICIT HumanString OPTIONAL,
  recordCount                 [14]     CHOICE {
                                           actualNumber    [0] IMPLICIT INTEGER,
                                           approxNumber    [1] IMPLICIT INTEGER} OPTIONAL,
```

```
  defaultOrder             [15]    IMPLICIT HumanString OPTIONAL,
  avRecordSize             [16]    IMPLICIT INTEGER OPTIONAL,
  maxRecordSize            [17]    IMPLICIT INTEGER OPTIONAL,
  hours                    [18]    IMPLICIT HumanString OPTIONAL,
  bestTime                 [19]    IMPLICIT HumanString OPTIONAL,
  lastUpdate               [20]    IMPLICIT GeneralizedTime OPTIONAL,
  updateInterval           [21]    IMPLICIT IntUnit OPTIONAL,
  coverage                 [22]    IMPLICIT HumanString OPTIONAL,
  proprietary              [23]    IMPLICIT BOOLEAN OPTIONAL, -- mandatory in full record
  copyrightText            [24]    IMPLICIT HumanString OPTIONAL,
  copyrightNotice          [25]    IMPLICIT HumanString OPTIONAL,
  producerContactInfo      [26]    IMPLICIT ContactInfo OPTIONAL,
  supplierContactInfo      [27]    IMPLICIT ContactInfo OPTIONAL,
  submissionContactInfo    [28]    IMPLICIT ContactInfo OPTIONAL,
    -- accessInfo lists items connected with the database. All listed items should be in the target's AccessInfo.
  accessInfo               [29]    IMPLICIT AccessInfo OPTIONAL}


SchemaInfo ::= SEQUENCE {
  commonInfo        [0]  IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
  schema            [1]  IMPLICIT OBJECT IDENTIFIER,
    -- Non-key brief elements follow:
  name              [2]  IMPLICIT InternationalString,
    -- Non-brief elements follow:
  description       [3]  IMPLICIT HumanString OPTIONAL,
  tagTypeMapping    [4]  IMPLICIT SEQUENCE OF SEQUENCE {
                           tagType          [0] IMPLICIT INTEGER,
                           tagSet           [1] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
                             -- If tagSet is omitted, then this tagType is for a tagSet locally defined
                             -- within the schema that cannot be referenced by another schema.
                           defaultTagType   [2] IMPLICIT NULL OPTIONAL
                                                  } OPTIONAL,
  recordStructure   [5]  IMPLICIT SEQUENCE OF ElementInfo OPTIONAL}


    --  ElementInfo referenced in SchemaInfo and RecordSyntaxInfo
        ElementInfo ::= SEQUENCE {
            elementName      [1] IMPLICIT InternationalString,
            elementTagPath   [2] IMPLICIT Path,
            dataType         [3] ElementDataType OPTIONAL, -- If omitted, not specified.
            required         [4] IMPLICIT BOOLEAN,
            repeatable       [5] IMPLICIT BOOLEAN,
            description      [6] IMPLICIT HumanString OPTIONAL}


    -- Path is referenced by ElementInfo as well as PerElementDetails
                Path ::= SEQUENCE OF SEQUENCE{
                tagType       [1] IMPLICIT INTEGER,
                tagValue      [2] StringOrNumeric}
            ElementDataType ::= CHOICE{
                primitive     [0] IMPLICIT PrimitiveDataType,
                structured    [1] IMPLICIT SEQUENCE OF ElementInfo}
```

```
        PrimitiveDataType ::= INTEGER{
            octetString        (0),
            numeric            (1),
            date               (2),
            external           (3),
            string             (4),
            trueOrFalse        (5),
            oid                (6),
            intUnit            (7),
            empty              (8),
            noneOfTheAbove     (100) -- see 'description'
                    }


TagSetInfo ::= SEQUENCE {
  commonInfo    [0]  IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
  tagSet        [1]  IMPLICIT OBJECT IDENTIFIER,
    -- non-key brief elements follow:
  name          [2]  IMPLICIT InternationalString,
    -- non-brief elements follow:
  description   [3]  IMPLICIT HumanString OPTIONAL,
  elements      [4]  IMPLICIT SEQUENCE OF SEQUENCE {
                       elementname     [1]  IMPLICIT InternationalString,
                       nicknames       [2]  IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
                       elementTag      [3]  StringOrNumeric,
                       description     [4]  IMPLICIT HumanString OPTIONAL,
                       dataType        [5]  PrimitiveDataType OPTIONAL,
                         -- If the data type is expected to be structured, that is described in the schema info,
                         -- and datatype is omitted here.
                       otherTagInfo         OtherInformation OPTIONAL} OPTIONAL}

RecordSyntaxInfo ::= SEQUENCE {
  commonInfo          [0]  IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
  recordSyntax        [1]  IMPLICIT OBJECT IDENTIFIER,
    -- Non-key brief elements follow:
  name                [2]  IMPLICIT InternationalString,
    -- non-brief elements follow:
  transferSyntaxes    [3]  IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
  description         [4]  IMPLICIT HumanString OPTIONAL,
  asn1Module          [5]  IMPLICIT InternationalString OPTIONAL,
  abstractStructure   [6]  IMPLICIT SEQUENCE OF ElementInfo OPTIONAL
                             -- Omitting abstractStructure only means target isn't using
                             -- Explain to describe the structure, not that there is no structure.
                       }

AttributeSetInfo ::= SEQUENCE {
  commonInfo    [0]  IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
  attributeSet  [1]  IMPLICIT AttributeSetId,
    -- non-key brief elements follow:
  name          [2]  IMPLICIT InternationalString,
```

```
    -- non-brief elements follow:
 attributes       [3]  IMPLICIT SEQUENCE OF AttributeType OPTIONAL,
                            -- mandatory in full record
 description      [4] IMPLICIT HumanString OPTIONAL}
-- AttributeType referenced in AttributeSetInfo
    AttributeType ::= SEQUENCE {
            name              [0] IMPLICIT InternationalString OPTIONAL,
            description       [1] IMPLICIT HumanString OPTIONAL,
            attributeType     [2] IMPLICIT INTEGER,
            attributeValues   [3] IMPLICIT SEQUENCE OF AttributeDescription}
        AttributeDescription ::= SEQUENCE {
                name                   [0] IMPLICIT InternationalString OPTIONAL,
                description            [1] IMPLICIT HumanString OPTIONAL,
                attributeValue         [2] StringOrNumeric,
                equivalentAttributes   [3]  IMPLICIT SEQUENCE OF StringOrNumeric OPTIONAL
                                       -- each is an occurrence of 'attributeValue' from AttributeDescription for a
                                       -- different attribute. Equivalences listed here should be derived from the
                                       -- attribute set definition, not from a particular server's behavior.
                                           }


TermListInfo ::= SEQUENCE{
 commonInfo       [0]  IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
 databaseName  [1]  IMPLICIT DatabaseName,
    -- Non-key brief elements follow:
 termLists        [2] IMPLICIT SEQUENCE OF SEQUENCE{
                    name         [1] IMPLICIT InternationalString,
                    title        [2] IMPLICIT HumanString OPTIONAL,
                                        -- Title is for users to see and can differ by language. Name, on the
                                        -- other hand is typically a short string not necessarily meant to be
                                        -- human-readable, and not variable by language.
                    searchCost   [3]   IMPLICIT INTEGER {
                                 optimized    (0),      -- The attribute (or combination) associated
                                                        -- with this    list will do fast searches.
                                 normal       (1),      -- The attribute (combination) will work as
                                                        -- expected. So there's probably an index for the
                                                        -- attribute (combination) or some similar
                                                        -- mechanism.
                                 expensive    (2),      -- Can use the attribute (combination), but it
                                                        -- might not provide satisfactory results.
                                                        -- Probably there is no index, or post-
                                                        -- processing of records is required.
                                 filter       (3)       -- can't search with this attribute (combination) alone.
                                              } OPTIONAL,
                    scanable     [4]  IMPLICIT BOOLEAN, -- 'true' means this list can be scanned.
                    broader      [5] IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
                    narrower     [6] IMPLICIT SEQUENCE OF InternationalString OPTIONAL
                                      -- broader and narrower list alternative term lists related to this one.
                                      -- The term lists so listed should also be in this termLists structure.
                                              }
    -- no non-brief elements
                    }
```

```
ExtendedServicesInfo ::= SEQUENCE {
  commonInfo              [0]  IMPLICIT CommonInfo OPTIONAL,
     -- Key elements follow:
  type                    [1]  IMPLICIT OBJECT IDENTIFIER,
     -- Non-key brief elements follow:
  name                    [2]  IMPLICIT InternationalString OPTIONAL,
                                    -- should be supplied if privateType is 'true'
  privateType             [3]  IMPLICIT BOOLEAN,
  restrictionsApply       [5]  IMPLICIT BOOLEAN,  -- if 'true' see 'description'
  feeApply                [6]  IMPLICIT BOOLEAN,  -- if 'true' see 'description'
  available               [7]  IMPLICIT BOOLEAN,
  retentionSupported      [8]  IMPLICIT BOOLEAN,
  waitAction              [9]  IMPLICIT INTEGER{
                                    waitSupported         (1),
                                    waitAlways            (2),
                                    waitNotSupported      (3),
                                    depends               (4),
                                    notSaying             (5)},
  -- non-brief elements follow:
          -- To get brief plus 'description' use esn 'description'
  description             [10]      IMPLICIT HumanString OPTIONAL,
          -- to get above elements and 'specificExplain' use esn 'specificExplain'
  specificExplain         [11]      IMPLICIT EXTERNAL OPTIONAL,
                                         -- Use oid of specific ES, and select choice [3] 'explain'. Format
                                         -- to be developed in conjunction with the specific ES definition.
          -- to get all elements except 'specificExplain', use esn 'asn'
  esASN                   [12]      IMPLICIT InternationalString OPTIONAL -- the ASN.1 for this ES
                                    }

-- - - - - - - - - - - - - Detail records
 -- The detail records describe relationships among entities supported by the target. RetrievalRecordDetails describes
 -- the way that schema elements are mapped into record elements. This mapping may be different for each
 -- combination of database, schema, record syntax. The per-element details describe the default mapping.
 -- Origin-request re-tagging can change that mapping. When multiple databases are listed in a databaseNames
 -- element, the record applies equally to all of the listed databases.  This is unrelated to searching the databases
 -- together. AttributeDetails describes how databases can be searched. Each supported attribute is listed, and the
 -- allowable combinations can be described.


AttributeDetails ::= SEQUENCE {
  commonInfo                   [0]  IMPLICIT CommonInfo OPTIONAL,
     -- Key elements follow:
  databaseName                 [1]  IMPLICIT DatabaseName,
     -- Non-brief elements follow:
  attributesBySet              [2]  IMPLICIT SEQUENCE OF AttributeSetDetails OPTIONAL,
                                         -- mandatory in full record
  attributeCombinations        [3]  IMPLICIT AttributeCombinations OPTIONAL}


-- AttributeSetDetails referenced by AttributeDetails
     AttributeSetDetails ::= SEQUENCE {
          attributeSet           [0] IMPLICIT AttributeSetId,
          attributesByType       [1] IMPLICIT SEQUENCE OF AttributeTypeDetails }
```

```
    AttributeTypeDetails ::= SEQUENCE {
        attributeType           [0] IMPLICIT INTEGER,
        defaultIfOmitted        [1] IMPLICIT OmittedAttributeInterpretation OPTIONAL,
        attributeValues         [2] IMPLICIT SEQUENCE OF AttributeValue OPTIONAL }
                                    -- If no attributeValues are supplied, all values of this type are fully
                                    -- supported, and the descriptions in AttributeSetInfo are adequate.

      OmittedAttributeInterpretation ::= SEQUENCE {
        defaultValue            [0] StringOrNumeric OPTIONAL,
                                    -- A default value is listed if that's how the server works
        defaultDescription      [1] IMPLICIT HumanString OPTIONAL }
                                    -- The human-readable description should generally be provided.
                                    -- It is legal for both default elements to be missing, which
                                    -- means that the target will allow the attribute type to be
                                    -- omitted, but isn't saying what it will do.

      AttributeValue ::= SEQUENCE {
       value                [0]  StringOrNumeric,
       description          [1]  IMPLICIT HumanString OPTIONAL,
       subAttributes        [2]  IMPLICIT SEQUENCE OF StringOrNumeric OPTIONAL,
       superAttributes      [3]  IMPLICIT SEQUENCE OF StringOrNumeric OPTIONAL,
       partialSupport       [4]  IMPLICIT NULL OPTIONAL }
            -- partialSupport indicates that an attributeValue is accepted, but may not be processed in the
            -- "expected" way. One important reason for this is composite databases: in this case partialSupport
            -- may indicate that only some of the subDbs support the attribute, and others ignore it.


TermListDetails ::= SEQUENCE{  -- one for each termList in TermListInfo
  commonInfo          [0]  IMPLICIT CommonInfo OPTIONAL,
     -- Key elements follow:
  termListName        [1]  IMPLICIT InternationalString,
     -- Non-key elements (all non-brief) follow:
  description         [2]  IMPLICIT HumanString OPTIONAL,
  attributes          [3]  IMPLICIT AttributeCombinations OPTIONAL,
                           -- Pattern for attributes that hit this list. Mandatory in full record
  scanInfo            [4]  IMPLICIT SEQUENCE {
                               maxStepSize          [0] IMPLICIT INTEGER OPTIONAL,
                               collatingSequence    [1] IMPLICIT HumanString OPTIONAL,
                               increasing           [2] IMPLICIT BOOLEAN OPTIONAL} OPTIONAL,
                           -- Occurs only if list is scanable. If list is scanable and if scanInfo is omitted,
                           -- target doesn't consider these important.
  estNumberTerms      [5]  IMPLICIT INTEGER OPTIONAL,
  sampleTerms         [6]  IMPLICIT SEQUENCE OF Term OPTIONAL}


ElementSetDetails ::= SEQUENCE {
     -- ElementSetDetails describes the way that database records are mapped to record elements. This
     -- mapping may be different for each combination of database name and element set. The database record
     -- description is a schema, which may be private to the target. The schema's abstract record structure
     -- and tag sets provide the vocabulary for discussing record content; their presence in the Explain
     -- database does not imply support for complex retrieval specification.
```

```
  commonInfo           [0]  IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
  databaseName         [1]  IMPLICIT DatabaseName,
  elementSetName       [2]  IMPLICIT ElementSetName,
  recordSyntax         [3]  IMPLICIT OBJECT IDENTIFIER,
    -- Non-key Brief elements follow:
  schema               [4]  IMPLICIT OBJECT IDENTIFIER,
    -- Non-brief elements follow:
  description          [5]  IMPLICIT HumanString OPTIONAL,
  detailsPerElement    [6]  IMPLICIT SEQUENCE OF PerElementDetails OPTIONAL -- mandatory in full record
                            }


RetrievalRecordDetails ::= SEQUENCE {
  commonInfo           [0]  IMPLICIT CommonInfo OPTIONAL,
      -- Key elements follow:
  databaseName         [1]  IMPLICIT DatabaseName,
  schema               [2]  IMPLICIT OBJECT IDENTIFIER,
  recordSyntax         [3]  IMPLICIT OBJECT IDENTIFIER,
    -- Non-brief elements follow:
  description          [4]  IMPLICIT HumanString OPTIONAL,
  detailsPerElement    [5]  IMPLICIT SEQUENCE OF PerElementDetails OPTIONAL
                                 -- mandatory in full record
                       }


-- PerElementDetails is referenced in RetrievalRecordDetails and ElementSetDetails.
    PerElementDetails ::= SEQUENCE {
        name            [0]     IMPLICIT InternationalString OPTIONAL,
                                    -- If the name is omitted, the record syntax's name for this element
                                    -- is appropriate.
        recordTag       [1]     IMPLICIT RecordTag OPTIONAL,
                                    -- The record tag may be omitted if tags are inappropriate for the record
                                    -- syntax, or if the origin can be expected to know it for some other reason.
        schemaTags      [2]     IMPLICIT SEQUENCE OF Path OPTIONAL,
                                    -- The information from the listed schema elements is combined
                                    -- in some way to produce the data sent in the listed record tag. The
                                    -- 'contents' element below may describe the logic used.
        maxSize         [3]     IMPLICIT INTEGER OPTIONAL,
        minSize         [4]     IMPLICIT INTEGER OPTIONAL,
        avgSize         [5]     IMPLICIT INTEGER OPTIONAL,
        fixedSize       [6]     IMPLICIT INTEGER OPTIONAL,
        repeatable      [8]     IMPLICIT BOOLEAN,
        required        [9]     IMPLICIT BOOLEAN,
                                    -- 'required' really means that target will always supply the element.
        description     [12]    IMPLICIT HumanString OPTIONAL,
        contents        [13]    IMPLICIT HumanString OPTIONAL,
        billingInfo     [14]    IMPLICIT HumanString OPTIONAL,
        restrictions    [15]    IMPLICIT HumanString OPTIONAL,
        alternateNames  [16]    IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
        genericNames    [17]    IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
        searchAccess    [18]    IMPLICIT AttributeCombinations OPTIONAL }
```

```
          -- RecordTag referenced in PerElementDetails above
              RecordTag ::= SEQUENCE {
                  qualifier     [0] StringOrNumeric OPTIONAL,
                                         -- E.g. tag set for GRS-1
                  tagValue      [1] StringOrNumeric}

SortDetails ::= SEQUENCE {
 commonInfo          [0]  IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
 databaseName        [1]  IMPLICIT DatabaseName,
    -- No non-key brief elements
    -- Non-brief elements follow:
 sortKeys            [2]  IMPLICIT SEQUENCE OF SortKeyDetails OPTIONAL
                               -- mandatory in full record
                     }
    SortKeyDetails ::= SEQUENCE {
        description               [0]  IMPLICIT HumanString OPTIONAL,
        elementSpecifications     [1]  IMPLICIT SEQUENCE OF Specification OPTIONAL,
                                         -- each specification is a way of specifying this same sort key
        attributeSpecifications   [2]  IMPLICIT AttributeCombinations OPTIONAL,
                                         -- each combination is a way of specifying this same sort key
        sortType                  [3]  CHOICE {
                                         character       [0]  IMPLICIT NULL,
                                         numeric         [1]  IMPLICIT NULL,
                                         structured      [2]  IMPLICIT HumanString} OPTIONAL,
        caseSensitivity           [4]  IMPLICIT INTEGER {
                                         always      (0), -- always case-sensitive
                                         never       (1), -- never case-sensitive
                                         default-yes (2), -- case-sensitivity is as specified on request, and if not
                                                          -- specified, case-sensitive.
                                         default-no  (3)} -- case-sensitivity is as specified on request, and if not
                                                          -- specified, not case-sensitive.
                                                     OPTIONAL}

ProcessingInformation ::= SEQUENCE{
 commonInfo               [0]  IMPLICIT CommonInfo OPTIONAL,
      -- Key elements follow:
 databaseName             [1]  IMPLICIT DatabaseName,
 processingContext        [2]  IMPLICIT INTEGER {
                                 access               (0),     -- e.g. choosing databases
                                 search               (1),     -- e.g. "search strategies" or search forms
                                 retrieval            (2),     -- e.g. recommended element combinations
                                 record-presentation  (3),     -- display of retrieved records
                                 record-handling      (4)      -- handling (e.g. saving) of retrieved records
                  },
 name                     [3]  IMPLICIT InternationalString,
 oid                      [4]  IMPLICIT OBJECT IDENTIFIER,
    -- No non-key brief elements
    -- Non-brief elements follow:
 description              [5]  IMPLICIT HumanString OPTIONAL,
                                 -- use element set name 'description' to retrieve all except instructions.
 instructions             [6]  IMPLICIT EXTERNAL OPTIONAL -- mandatory in full record
                               }
```

```
VariantSetInfo ::= SEQUENCE {
        -- A record in this category describes a variant set definition, i.e., classes, types, and values, for a specific
        -- variant set definition supported by the target. Support by the target of a particular variant set definition
        -- does not imply that the definition is supported for any specific database or element.
 commonInfo          [0] IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
 variantSet       [1] IMPLICIT OBJECT IDENTIFIER,
    -- Non-key brief elements follow:
 name             [2] IMPLICIT InternationalString,
    -- Non-brief elements follow:
 variants         [3] IMPLICIT SEQUENCE OF VariantClass OPTIONAL
                        -- mandatory in full record
                         }

    -- Subsidiary structures for VariantSetInfo
        VariantClass ::= SEQUENCE {
            name              [0] IMPLICIT InternationalString OPTIONAL,
            description       [1] IMPLICIT HumanString OPTIONAL,
            variantClass      [2] IMPLICIT INTEGER,
            variantTypes      [3] IMPLICIT SEQUENCE OF VariantType}
        VariantType ::= SEQUENCE {
            name              [0] IMPLICIT InternationalString OPTIONAL,
            description       [1] IMPLICIT HumanString OPTIONAL,
            variantType       [2] IMPLICIT INTEGER,
            variantValue      [3] IMPLICIT VariantValue OPTIONAL}
        VariantValue ::= SEQUENCE {
            dataType          [0] PrimitiveDataType,
            values            [1] ValueSet OPTIONAL }
        ValueSet ::= CHOICE {
            range         [0] IMPLICIT ValueRange,
            enumerated    [1] IMPLICIT SEQUENCE OF ValueDescription }
        ValueRange ::= SEQUENCE {
                -- At last one the following must be supplied, both may be supplied.
            lower         [0] ValueDescription OPTIONAL,
            upper         [1] ValueDescription OPTIONAL }
        ValueDescription ::= CHOICE{
            integer                  INTEGER,
            string                   InternationalString,
            octets                   OCTET STRING,
            oid                      OBJECT IDENTIFIER,
            unit              [1]  IMPLICIT Unit,
            valueAndUnit      [2]  IMPLICIT IntUnit
                -- oid and unit can't be used in a ValueRange
                         }


UnitInfo ::= SEQUENCE {
 commonInfo      [0] IMPLICIT CommonInfo OPTIONAL,
    -- Key elements follow:
 unitSystem      [1] IMPLICIT InternationalString,
    -- No non-key brief elements
    -- Non-brief elements follow:
```

```
  description      [2] IMPLICIT HumanString OPTIONAL,
  units            [3] IMPLICIT SEQUENCE OF UnitType OPTIONAL
                             -- mandatory in full record
                         }

    -- Subsidiary structures for UnitInfo
         UnitType ::= SEQUENCE {
              name          [0] IMPLICIT InternationalString OPTIONAL,
              description   [1] IMPLICIT HumanString OPTIONAL,
              unitType      [2] StringOrNumeric,
              units         [3] IMPLICIT SEQUENCE OF Units}

         Units ::= SEQUENCE {
              name          [0] IMPLICIT InternationalString OPTIONAL,
              description   [1] IMPLICIT HumanString OPTIONAL,
              unit          [2] StringOrNumeric}

CategoryList ::= SEQUENCE {
  commonInfo       [0] IMPLICIT CommonInfo OPTIONAL,
    -- Only one record expected per Explain database. All elements appear in brief presentation.
  categories       [1] IMPLICIT SEQUENCE OF CategoryInfo }
         CategoryInfo ::= SEQUENCE {
              category              [1] IMPLICIT InternationalString,
              originalCategory      [2] IMPLICIT InternationalString OPTIONAL,
              description           [3] IMPLICIT HumanString OPTIONAL,
              asn1Module            [4] IMPLICIT InternationalString OPTIONAL}


-- - - - - - - - - - - - - - Subsidiary definitions

CommonInfo ::= SEQUENCE {
  dateAdded                [0] IMPLICIT GeneralizedTime OPTIONAL,
  dateChanged              [1] IMPLICIT GeneralizedTime OPTIONAL,
  expiry                   [2] IMPLICIT GeneralizedTime OPTIONAL,
  humanString-Language     [3] IMPLICIT LanguageCode OPTIONAL,
    -- following not to occur for brief:
  otherInfo                    OtherInformation OPTIONAL}

HumanString ::= SEQUENCE OF SEQUENCE {
                   language      [0] IMPLICIT LanguageCode OPTIONAL,
                   text          [1] IMPLICIT InternationalString}

IconObject ::= SEQUENCE OF SEQUENCE{
              -- Note that the "SEQUENCE OF" is to allow alternative representations of the same Icon; it is not
              -- intended to allow multiple icons.
    bodyType [1] CHOICE{
                        ianaType      [1] IMPLICIT InternationalString,
                        z3950type     [2] IMPLICIT InternationalString,
                        otherType     [3] IMPLICIT InternationalString},
    content  [2] IMPLICIT OCTET STRING}


LanguageCode ::= InternationalString  -- from ANSI/NISO Z39.53-1994
```

```
ContactInfo ::= SEQUENCE {
  name              [0] IMPLICIT InternationalString OPTIONAL,
  description       [1] IMPLICIT HumanString OPTIONAL,
  address           [2] IMPLICIT HumanString OPTIONAL,
  email             [3] IMPLICIT InternationalString OPTIONAL,
  phone             [4] IMPLICIT InternationalString OPTIONAL}

NetworkAddress ::= CHOICE {
  internetAddress             [0] IMPLICIT SEQUENCE {
                                   hostAddress     [0]  IMPLICIT InternationalString,
                                   port            [1]  IMPLICIT INTEGER},
  osiPresentationAddress      [1] IMPLICIT SEQUENCE {
                                   pSel            [0] IMPLICIT InternationalString,
                                   sSel            [1] IMPLICIT InternationalString OPTIONAL,
                                   tSel            [2] IMPLICIT InternationalString OPTIONAL,
                                   nSap            [3] IMPLICIT InternationalString},
  other                       [2] IMPLICIT SEQUENCE {
                                   type        [0] IMPLICIT InternationalString,
                                   address     [1] IMPLICIT InternationalString}}


AccessInfo ::= SEQUENCE {
        -- AccessInfo contains the fundamental information about what facilities are required to use this target
        -- or server. For example, if an origin can handle none of the record syntaxes a database can provide,
        -- it might choose not to access the database.
  queryTypesSupported    [0]     IMPLICIT SEQUENCE OF QueryTypeDetails OPTIONAL,
  diagnosticsSets        [1]     IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
  attributeSetIds        [2]     IMPLICIT SEQUENCE OF AttributeSetId OPTIONAL,
  schemas                [3]     IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
  recordSyntaxes         [4]     IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
  resourceChallenges     [5]     IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
  restrictedAccess       [6]     IMPLICIT AccessRestrictions OPTIONAL,
  costInfo               [8]     IMPLICIT Costs OPTIONAL,
  variantSets            [9]     IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
  elementSetNames        [10]    IMPLICIT SEQUENCE OF ElementSetName OPTIONAL,
  unitSystems            [11]    IMPLICIT SEQUENCE OF InternationalString}

-- begin auxiliary definitions for AccessInfo
-- Begin Query Details
QueryTypeDetails ::= CHOICE {
  private            [0]      IMPLICIT PrivateCapabilities,
  rpn                [1]      IMPLICIT RpnCapabilities,
  iso8777            [2]      IMPLICIT Iso8777Capabilities,
  z39-58             [100]    IMPLICIT HumanString,
  erpn               [101]    IMPLICIT RpnCapabilities,
  rankedList         [102]    IMPLICIT HumanString}

PrivateCapabilities ::= SEQUENCE {
  operators        [0]  IMPLICIT SEQUENCE OF SEQUENCE {
                             operator       [0] IMPLICIT InternationalString,
                             description    [1] IMPLICIT HumanString OPTIONAL } OPTIONAL,
  searchKeys       [1]  IMPLICIT SEQUENCE OF SearchKey OPTIONAL,  -- field names that can be searched
  description      [2]  IMPLICIT SEQUENCE OF HumanString OPTIONAL }
```

```
RpnCapabilities ::= SEQUENCE {
  operators                        [0] IMPLICIT SEQUENCE OF INTEGER OPTIONAL,
                                               -- Omitted means all operators are supported.
  resultSetAsOperandSupported      [1] IMPLICIT BOOLEAN,
  restrictionOperandSupported      [2] IMPLICIT BOOLEAN,
  proximity                        [3] IMPLICIT ProximitySupport OPTIONAL}

Iso8777Capabilities ::= SEQUENCE {
  searchKeys      [0]  IMPLICIT SEQUENCE OF SearchKey,  -- field names that may be searched
  restrictions    [1]  IMPLICIT HumanString OPTIONAL
                           -- Omitted means supported, not specifying units.
                               }

ProximitySupport ::= SEQUENCE {
  anySupport      [0]   IMPLICIT BOOLEAN,
                       -- 'false' means no proximity support, in which case unitsSupported not supplied.
  unitsSupported  [1]   IMPLICIT SEQUENCE OF CHOICE{
                       known   [1] IMPLICIT INTEGER, -- values from KnownProximityUnit
                       private [2] IMPLICIT SEQUENCE{
                                   unit          [0] IMPLICIT INTEGER,
                                   description   [1] HumanString OPTIONAL}} OPTIONAL}

SearchKey ::= SEQUENCE {
  searchKey       [0]  IMPLICIT InternationalString,
  description     [1]  IMPLICIT HumanString OPTIONAL }
-- End Query details

AccessRestrictions ::= SEQUENCE OF SEQUENCE {
  accessType           [0] INTEGER {
                           any                    (0),
                           search                 (1),
                           present                (2),
                           specific-elements      (3),
                           extended-services      (4),
                           by-database            (5)},
  accessText           [1]  IMPLICIT HumanString OPTIONAL,
  accessChallenges     [2]  IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL}

Costs ::= SEQUENCE {
  connectCharge        [0] IMPLICIT Charge OPTIONAL,     -- Per-connection charge
  connectTime          [1] IMPLICIT Charge OPTIONAL,     -- Time-based charge
  displayCharge        [2] IMPLICIT Charge OPTIONAL,     -- Per-record charge
  searchCharge         [3] IMPLICIT Charge OPTIONAL,     -- Per-search charge
  subscriptCharge      [4]  IMPLICIT Charge OPTIONAL,    -- Subscription charges
  otherCharges         [5] IMPLICIT SEQUENCE OF SEQUENCE{   -- Other charges
                           forWhat     [1]  IMPLICIT HumanString,
                           charge      [2]  IMPLICIT Charge} OPTIONAL}
        Charge ::= SEQUENCE{
              cost         [1] IMPLICIT IntUnit,
              perWhat      [2] IMPLICIT Unit OPTIONAL,
                           -- e.g. "second," "minute," "line," "record"...
              text         [3] IMPLICIT HumanString OPTIONAL}
-- End Auxiliary definitions for AccessInfo
```

```
DatabaseList ::= SEQUENCE OF DatabaseName

AttributeCombinations ::= SEQUENCE {
  defaultAttributeSet  [0] IMPLICIT AttributeSetId,
                                -- Default for the combinations. Also probably a good choice for the default
                                -- in searches, but that isn't required.
  legalCombinations   [1] IMPLICIT SEQUENCE OF AttributeCombination }


AttributeCombination ::= SEQUENCE OF AttributeOccurrence
                         -- An AttributeCombination is a pattern for legal combination of attributes


AttributeOccurrence ::= SEQUENCE {
          -- An AttributeOccurrence lists the legal values for a specific attribute type in a combination.
     attributeSet        [0]  IMPLICIT AttributeSetId OPTIONAL,
     attributeType       [1]  IMPLICIT INTEGER,
     mustBeSupplied      [2]  IMPLICIT NULL OPTIONAL,
     attributeValues     CHOICE {
                              any-or-none  [3] IMPLICIT NULL, -- All supported values are OK
                              specific     [4] IMPLICIT SEQUENCE OF StringOrNumeric}}
                                                -- Only these values allowed
END
```

REC.2 Simple Unstructured Text Record Syntax

The Simple Unstructured Text Record Syntax (SUTRS) is intended to be used as a record syntax in a Search or Present response, to present textual data so that the origin may display them with little or no analysis and manipulation. A SUTRS record is unstructured; the text of a SUTRS record might represent individual elements, but the elements are not explicitly identified by the syntax. The convention prescribed by the SUTRS definition is to use a delimiter within the text to indicate the end of a line of text. The prescribed line terminator is ASCII LF (X'0A'). Thus a SUTRS record consists simply of a string of textual data.

This definition recommends that the maximum line length be 72 characters unless an alternative maximum is requested, for example via a variantRequest. This is not an absolute maximum, but it is recommended that targets make a best effort to limit lines to this length.

RecordSyntax-SUTRS

```
{Z39-50-recordSyntax SUTRS (101)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString  FROM Z39-50-APDU-1995;
        SutrsRecord ::=  InternationalString
-- Line terminator is ASCII LF (X'0A').
-- Recommended maximum line length is 72 characters.
END
```

REC.3 OPAC Record Syntax

```
RecordSyntax-opac
{Z39-50-recordSyntax opac (102)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString  FROM Z39-50-APDU-1995;
OPACRecord ::= SEQUENCE {
          bibliographicRecord      [1]     IMPLICIT EXTERNAL OPTIONAL,
          holdingsData             [2]     IMPLICIT SEQUENCE OF HoldingsRecord OPTIONAL}
HoldingsRecord ::=  CHOICE {
          marcHoldingsRecord       [1]     IMPLICIT EXTERNAL,
          holdingsAndCirc          [2]     IMPLICIT HoldingsAndCircData}
HoldingsAndCircData ::=  SEQUENCE {
-- the following elements are required to display holdings in conformance with NISO standards.
          typeOfRecord             [1]     IMPLICIT InternationalString OPTIONAL, -- LDR 06
          encodingLevel            [2]     IMPLICIT InternationalString OPTIONAL, -- LDR 017
          format                   [3]     IMPLICIT InternationalString OPTIONAL, -- 007 00-01
          receiptAcqStatus         [4]     IMPLICIT InternationalString OPTIONAL, -- 008 06
          generalRetention         [5]     IMPLICIT InternationalString OPTIONAL, -- 008 12
          completeness             [6]     IMPLICIT InternationalString OPTIONAL, -- 008 16
          dateOfReport             [7]     IMPLICIT InternationalString OPTIONAL, -- 008 26-31
          nucCode                  [8]     IMPLICIT InternationalString OPTIONAL, -- 852 $a
          localLocation            [9]     IMPLICIT InternationalString OPTIONAL, -- 852 $b
          shelvingLocation         [10]    IMPLICIT InternationalString OPTIONAL, -- 852 $c
          callNumber               [11]    IMPLICIT InternationalString OPTIONAL, -- 852 $h and $i
          shelvingData             [12]    IMPLICIT InternationalString OPTIONAL, -- 852 $j thru $m
          copyNumber               [13]    IMPLICIT InternationalString OPTIONAL, -- 852 $t
          publicNote               [14]    IMPLICIT InternationalString OPTIONAL, -- 852 $z
          reproductionNote         [15]    IMPLICIT InternationalString OPTIONAL, -- 843
          termsUseRepro            [16]    IMPLICIT InternationalString OPTIONAL, -- 845
          enumAndChron             [17]    IMPLICIT InternationalString OPTIONAL, -- all 85x, 86x
          volumes                  [18]    IMPLICIT SEQUENCE OF Volume OPTIONAL,
                                                     -- repeats for each volume held
          circulationData          [19]    IMPLICIT SEQUENCE OF CircRecord OPTIONAL
                                                     -- repeats for each circulating item.
                                                     }
Volume ::=  SEQUENCE {
          enumeration              [1]     IMPLICIT InternationalString OPTIONAL,
          chronology               [2]     IMPLICIT InternationalString OPTIONAL,
          enumAndChron             [3]     IMPLICIT InternationalString OPTIONAL }
CircRecord ::=  SEQUENCE {
          availableNow             [1]     IMPLICIT BOOLEAN,
          availablityDate          [2]     IMPLICIT InternationalString OPTIONAL,
          availableThru            [3]     IMPLICIT InternationalString OPTIONAL,
          restrictions             [4]     IMPLICIT InternationalString OPTIONAL,
          itemId                   [5]     IMPLICIT InternationalString OPTIONAL,
          renewable                [6]     IMPLICIT BOOLEAN,
          onHold                   [7]     IMPLICIT BOOLEAN,
          enumAndChron             [8]     IMPLICIT InternationalString OPTIONAL,
          midspine                 [9]     IMPLICIT InternationalString OPTIONAL,
          temporaryLocation        [10]    IMPLICIT InternationalString OPTIONAL}
END
```

REC.4 Summary Record Syntax

RecordSyntax-summary

```
{Z39-50-recordSyntax summary (103)} DEFINITIONS ::=
BEGIN
IMPORTS OtherInformation, InternationalString  FROM Z39-50-APDU-1995;
BriefBib ::= SEQUENCE {
        title                   [1]     IMPLICIT InternationalString,
        author                  [2]     IMPLICIT InternationalString OPTIONAL,
        callNumber              [3]     IMPLICIT InternationalString OPTIONAL,
        recordType              [4]     IMPLICIT InternationalString OPTIONAL,
        bibliographicLevel      [5]     IMPLICIT InternationalString OPTIONAL,
        format                  [6]     IMPLICIT SEQUENCE OF FormatSpec OPTIONAL,
        publicationPlace        [7]     IMPLICIT InternationalString OPTIONAL,
        publicationDate         [8]     IMPLICIT InternationalString OPTIONAL,
        targetSystemKey         [9]     IMPLICIT InternationalString OPTIONAL,
        satisfyingElement       [10]    IMPLICIT InternationalString OPTIONAL,
        rank                    [11]    IMPLICIT INTEGER OPTIONAL,
        documentId              [12]    IMPLICIT InternationalString OPTIONAL,
        abstract                [13]    IMPLICIT InternationalString OPTIONAL,
        otherInfo                       OtherInformation OPTIONAL}

FormatSpec    ::=  SEQUENCE {
        type            [1]     IMPLICIT InternationalString,
        size            [2]     IMPLICIT INTEGER OPTIONAL,
        bestPosn        [3]     IMPLICIT INTEGER OPTIONAL}
END
```

REC.5 Generic Record Syntax 1

RecordSyntax-generic -- *For detailed semantics, see Appendix RET.*

```
{Z39-50-recordSyntax GRS-1 (105)} DEFINITIONS ::=
BEGIN
EXPORTS Variant;
IMPORTS IntUnit, Unit, InternationalString, StringOrNumeric, Term FROM Z39-50-APDU-1995;

GenericRecord ::= SEQUENCE OF TaggedElement
TaggedElement ::= SEQUENCE {
    tagType          [1] IMPLICIT INTEGER OPTIONAL,
                         -- If omitted, default should be supplied dynamically by tagSet-M;
                         -- otherwise it should be statically specified by the schema.
    tagValue         [2] StringOrNumeric,
    tagOccurrence    [3] IMPLICIT INTEGER OPTIONAL,
                         -- Occurrence within the database record, and relative to the parent. No
                         -- default; if omitted, target not telling or it is irrelevant.
    content          [4] ElementData,
    metaData         [5] IMPLICIT ElementMetaData OPTIONAL,
    appliedVariant   [6] IMPLICIT Variant OPTIONAL}
```

```
ElementData ::= CHOICE{
     octets              OCTET STRING,
     numeric             INTEGER,
     date                GeneralizedTime,
     ext                 EXTERNAL,
     string              InternationalString,
     trueOrFalse         BOOLEAN,
     oid                 OBJECT IDENTIFIER,
     intUnit                     [1]  IMPLICIT IntUnit,
     elementNotThere             [2] IMPLICIT NULL,  -- element requested but not there
     elementEmpty                [3] IMPLICIT NULL,  -- element there, but empty
     noDataRequested             [4] IMPLICIT NULL,  -- variant request said 'no data'
     diagnostic                  [5] IMPLICIT EXTERNAL,
     subtree                     [6] SEQUENCE OF TaggedElement -- recursive, for nested tags
                                 }


ElementMetaData ::= SEQUENCE{
     seriesOrder                 [1]     IMPLICIT Order OPTIONAL, -- only for a non-leaf node
     usageRight                  [2]     IMPLICIT Usage OPTIONAL,
     hits                        [3]     IMPLICIT SEQUENCE OF HitVector OPTIONAL,
     displayName                 [4]     IMPLICIT InternationalString OPTIONAL,
                                               -- name for element that origin can use for display
     supportedVariants           [5]     IMPLICIT SEQUENCE OF Variant OPTIONAL,
     message                     [6]     IMPLICIT InternationalString OPTIONAL,
     elementDescriptor           [7]     IMPLICIT OCTET STRING OPTIONAL,
     surrogateFor                [8]     IMPLICIT TagPath OPTIONAL,
                                               -- the retrieved element is a surrogate for the element given by this path
     surrogateElement            [9]     IMPLICIT TagPath OPTIONAL,
                                               -- the element given by this path is a surrogate for the retrieved element
     other                       [99]    IMPLICIT EXTERNAL OPTIONAL}
               TagPath ::= SEQUENCE  OF SEQUENCE{
                    tagType                 [1]  IMPLICIT INTEGER OPTIONAL,
                    tagValue                [2]  StringOrNumeric,
                    tagOccurrence           [3]  IMPLICIT INTEGER OPTIONAL}


Order ::= SEQUENCE{
     ascending           [1] IMPLICIT BOOLEAN,
                                 -- "true" means monotonically increasing (i.e. non-decreasing);
                                 -- "false" means monotonically decreasing (i.e. non-increasing).
     order               [2] IMPLICIT INTEGER
                                 -- Same as defined by 'elementOrdering' in tagSet-M, though this may be
                                 -- overridden by schema.
                         }

Usage ::= SEQUENCE    {
     type                [1] IMPLICIT INTEGER{
                                 redistributable          (1),      -- Element is freely redistributable.
                                 restricted               (2),      -- Restriction contains statement.
                                 licensePointer           (3)       -- Restriction contains license pointer.
                                 },
     restriction         [2] IMPLICIT InternationalString OPTIONAL}
```

```
HitVector ::= SEQUENCE{
        -- Each hit vector points to a fragment within the element, via location and/or token.
    satisfier                   Term OPTIONAL, -- sourceword, etc.
    offsetIntoElement       [1] IMPLICIT IntUnit OPTIONAL,
    length                  [2] IMPLICIT IntUnit OPTIONAL,
    hitRank                 [3] IMPLICIT INTEGER OPTIONAL,
    targetToken             [4] IMPLICIT OCTET STRING OPTIONAL
                                -- Origin may use token subsequently within a variantRequest (in
                                --  an elementRequest) to retrieve (or to refer to) the fragment.
                            }


Variant ::= SEQUENCE{
    globalVariantSetId      [1] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
                                -- Applies to the triples below, when variantSetId omitted. If
                                -- globalVariantSetId omitted, default applies. Default may be provided by
                                -- the tagSet-M element defaultVariantSetId.
    triples                 [2]  IMPLICIT SEQUENCE OF SEQUENCE{
                                variantSetId  [0]  IMPLICIT OBJECT IDENTIFIER OPTIONAL,
                                                   -- If omitted, globalVariantSetId (above)
                                                   -- applies, unless that too is omitted, in
                                                   -- which case, default used.
                                class              [1]  IMPLICIT INTEGER,
                                type               [2]  IMPLICIT INTEGER,
                                value              [3]  CHOICE{
                                                        INTEGER,
                                                        InternationalString,
                                                        OCTET STRING,
                                                        OBJECT IDENTIFIER,
                                                        BOOLEAN,
                                                        NULL,
                                                   -- Following need context tags:
                                                     unit            [1] IMPLICIT Unit,
                                                     valueAndUnit    [2] IMPLICIT IntUnit}}}
END
```

REC.6 Record Syntax For Extended Services Task Package

```
RecordSyntax-ESTaskPackage
{Z39-50-recordSyntax ESTaskPackage (106)} DEFINITIONS ::=
BEGIN
IMPORTS Permissions, InternationalString, IntUnit, DiagRec FROM Z39-50-APDU-1995;

TaskPackage ::= SEQUENCE{
    packageType              [1]    IMPLICIT OBJECT IDENTIFIER,
                                        -- oid of specific ES definition
    packageName              [2]    IMPLICIT InternationalString OPTIONAL,
    userId                   [3]    IMPLICIT InternationalString OPTIONAL,
    retentionTime            [4]    IMPLICIT IntUnit OPTIONAL,
    permissions              [5]    IMPLICIT Permissions OPTIONAL,
    description              [6]    IMPLICIT InternationalString OPTIONAL,
    targetReference          [7]    IMPLICIT OCTET STRING OPTIONAL,
    creationDateTime         [8]    IMPLICIT GeneralizedTime OPTIONAL,
    taskStatus               [9]    IMPLICIT INTEGER{
                                        pending      (0),
                                        active       (1),
                                        complete     (2),
                                        aborted      (3)},
    packageDiagnostics       [10]   IMPLICIT SEQUENCE OF DiagRec OPTIONAL,
    taskSpecificParameters   [11]   IMPLICIT EXTERNAL
                                        -- Use oid for specific ES definition
                                        -- (same oid as packageType above)
                                        -- and select [2] "taskPackage."

        }
END
```