Appendix 18 ASN1: Z39.50 ASN.1

Normative

ASN1.1 Z39.50 APDUs

In this version (Z39.50-2001) some of the longer comments are presented outside of the actual ASN.1 module below, and for each of these there is a reference to the comment, within the ASN.1 at the point where the comment applies. All such comments, as well as comments included within the actual ASN.1, are normative and are part of the standard.

```
Z39-50-APDU-2001 -OID for this definition, assigned in OID.3.1, is {Z39-50 2 1}
DEFINITIONS ::=
BEGIN  -- November 2001
EXPORTS OtherInformation, Term, AttributeSetId, AttributeList, AttributeElement, ElementSetName,
SortElement, DatabaseName, CompSpec, Specification, Permissions, InternationalString, IntUnit, Unit,
StringOrNumeric, Query, Records, ResultSetId, DefaultDiagFormat, DiagRec;
APDU ::= CHOICE{
        initRequest                     [20] IMPLICIT InitializeRequest,
        initResponse                    [21] IMPLICIT InitializeResponse,
        searchRequest                   [22] IMPLICIT SearchRequest,
        searchResponse                  [23] IMPLICIT SearchResponse,
        presentRequest                  [24] IMPLICIT PresentRequest,
        presentResponse                 [25] IMPLICIT PresentResponse,
        deleteResultSetRequest          [26] IMPLICIT DeleteResultSetRequest,
        deleteResultSetResponse         [27] IMPLICIT DeleteResultSetResponse,
        accessControlRequest            [28] IMPLICIT AcessControlRequest,
        accessControlResponse           [29] IMPLICIT AccessControlResponse,
        resourceControlRequest          [30] IMPLICIT ResourceControlRequest,
        resourceControlResponse         [31] IMPLICIT ResourceControlResponse,
        triggerResourceControlRequest   [32] IMPLICIT TriggerResourceControlRequest,
        resourceReportRequest           [33] IMPLICIT ResourceReportRequest,
        resourceReportResponse          [34] IMPLICIT ResourceReportResponse,
        scanRequest                     [35] IMPLICIT ScanRequest,
        scanResponse                    [36] IMPLICIT ScanResponse,
                        -- [37] through [42] not used
        sortRequest                     [43] IMPLICIT SortRequest,
        sortResponse                    [44] IMPLICIT SortResponse,
        segmentRequest                  [45] IMPLICIT Segment,
        extendedServicesRequest         [46] IMPLICIT ExtendedServicesRequest,
        extendedServicesResponse        [47] IMPLICIT ExtendedServicesResponse,
        close                           [48] IMPLICIT Close,
        duplicateDetectionRequest       [49] IMPLICIT ExtendedServicesRequest,
        duplicateDetectionResponse      [50] IMPLICIT DuplicateDetectionResponse}

-- Initialize APDUs
InitializeRequest ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        protocolVersion                         ProtocolVersion,
        options                                 Options,
        preferredMessageSize            [5]     IMPLICIT INTEGER,
        exceptionalRecordSize           [6]     IMPLICIT INTEGER,
        idAuthentication                [7]     ANY OPTIONAL,
```

```
                -- See comment 8
        implementationId                [110]   IMPLICIT InternationalString OPTIONAL,
        implementationName              [111]   IMPLICIT InternationalString OPTIONAL,
        implementationVersion           [112]   IMPLICIT InternationalString OPTIONAL,
        userInformationField            [11]    EXTERNAL OPTIONAL,
        otherInfo                               OtherInformation OPTIONAL}

InitializeResponse ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        protocolVersion                         ProtocolVersion,
        options                                 Options,
        preferredMessageSize            [5]     IMPLICIT INTEGER,
        exceptionalRecordSize           [6]     IMPLICIT INTEGER,
                result                  [12]    IMPLICIT BOOLEAN,
                        -- reject = FALSE; Accept = TRUE
        implementationId                [110]   IMPLICIT InternationalString OPTIONAL,
        implementationName              [111]   IMPLICIT InternationalString OPTIONAL,
        implementationVersion           [112]   IMPLICIT InternationalString OPTIONAL,
        userInformationField            [11]    EXTERNAL OPTIONAL,
        otherInfo                               OtherInformation OPTIONAL}

    --Begin auxiliary definitions for Init APDUs

ProtocolVersion  ::=[3] IMPLICIT BIT STRING{
        version-1       (0),            -- This bit should always be set,
                                        --but does not correspond to any Z39.50 version.
        version-2       (1),            -- "Version 2 supported." This bit should always be set.
        version-3       (2)             -- "Version 3 supported."
    -- See comment 9
    }

Options  ::= [4] IMPLICIT BIT STRING{
        search                  (0),
        present                 (1),
        delSet                  (2),
        resourceReport          (3),
        triggerResourceCtrl     (4),
        resourceCtrl            (5),
        accessCtrl              (6),
        scan                    (7),
        sort                    (8),
                -- (not used)   (9),
        extendedServices        (10),
        level-1Segmentation     (11),
        level-2Segmentation     (12),
        concurrentOperations    (13),
        namedResultSets         (14),
        Encapsulation           (15),
        resultCountInSort       (16),
        negotiation             (17),
        Dedup                   (18),
        query104                (19),
        PQESCorrection          (20),
        stringSchema            (21)}
-- End auxiliary definitions for Init APDUs
```

```
-- Search APDUs
SearchRequest ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        smallSetUpperBound              [13]    IMPLICIT INTEGER,
        largeSetLowerBound              [14]    IMPLICIT INTEGER,
        mediumSetPresentNumber          [15]    IMPLICIT INTEGER,
        replaceIndicator                [16]    IMPLICIT BOOLEAN,
        resultSetName                   [17]    IMPLICIT InternationalString,
        databaseNames                   [18]    IMPLICIT SEQUENCE OF DatabaseName,
        smallSetElementSetNames         [100]   ElementSetNames OPTIONAL,
        mediumSetElementSetNames        [101]   ElementSetNames OPTIONAL,
        preferredRecordSyntax           [104]   IMPLICIT OBJECT IDENTIFIER OPTIONAL,
        query                           [21]    Query,
-- Following two parameters may be used only if version 3 is in force.
        additionalSearchInfo            [203]   IMPLICIT OtherInformation OPTIONAL,
        otherInfo                               OtherInformation OPTIONAL}

--Query Definitions
Query  ::=          CHOICE{
        type-0  [0]     ANY,
        type-1  [1]     IMPLICIT RPNQuery,
        type-2  [2]     OCTET STRING,
        type-100[100]   OCTET STRING,
        type-101[101]   IMPLICIT RPNQuery,
        type-102[102]   OCTET STRING,
        type-104[104]   IMPLICIT EXTERNAL}

--Definitions for RPN query
RPNQuery ::= SEQUENCE{
        attributeSet    AttributeSetId,
        rpn             RPNStructure}
RPNStructure ::= CHOICE{
        op              [0] Operand,
        rpnRpnOp        [1] IMPLICIT SEQUENCE{
                        rpn1                    RPNStructure,
                        rpn2                    RPNStructure,
                        op                      Operator }}
Operand ::= CHOICE{
        attrTerm        AttributesPlusTerm,
        resultSet       ResultSetId,
--If version 2 is in force:
--If query type is 1, one of the above two must be chosen
--resultAttr (below) may be used only if query type is 101
resultAttr              ResultSetPlusAttributes}
AttributesPlusTerm ::= [102] IMPLICIT SEQUENCE{
        attributes              AttributeList,
        term                    Term}
ResultSetPlusAttributes ::= [214] IMPLICIT SEQUENCE{
        resultSet       ResultSetId,
        attributes      AttributeList}
AttributeList ::=       [44]    IMPLICIT SEQUENCE OF AttributeElement
Term ::= CHOICE{
        general                 [45]    IMPLICIT OCTET STRING,
-- Values below may be used only if version 3 is in force
```

```
        numeric                 [215]   IMPLICIT INTEGER,
        characterString         [216]   IMPLICIT InternationalString,
        oid                     [217]   IMPLICIT OBJECT IDENTIFIER,
        dateTime                [218]   IMPLICIT GeneralizedTime,
        external                [219]   IMPLICIT EXTERNAL,
        integerAndUnit          [220]   IMPLICIT IntUnit,
        null                    [221]   IMPLICIT NULL}
Operator ::= [46] CHOICE{
        and                     [0] IMPLICIT NULL,
        or                      [1] IMPLICIT NULL,
        and-not                 [2] IMPLICIT NULL,
--If version 2 is in force:
--For query type 1, one of the above three must be chosen;
--prox (below) may be used only if query type is 101.
        prox                    [3] IMPLICIT ProximityOperator}
AttributeElement ::= SEQUENCE{
        attributeSet    [1]     IMPLICIT AttributeSetId OPTIONAL,
--Must be omitted if version 2 is in force.
--If included, overrides value of attributeSet in RPNQuery above, but only for this attribute.
        attributeType   [120]   IMPLICIT INTEGER,
        attributeValue          CHOICE{
                                numeric  [121] IMPLICIT INTEGER,
                -- If version 2 is in force, must select 'numeric' for attributeValue
                                complex [224] IMPLICIT SEQUENCE{
                                        list    [1]  IMPLICIT SEQUENCE OF StringOrNumeric,
                                semanticAction  [2]  IMPLICIT SEQUENCE OF INTEGER
                                                        OPTIONAL}
        -- See comment 10.
        }}
ProximityOperator ::= SEQUENCE{
        exclusion               [1] IMPLICIT BOOLEAN OPTIONAL,
        distance                [2] IMPLICIT INTEGER,
        ordered                 [3] IMPLICIT BOOLEAN,
        relationType            [4] IMPLICIT INTEGER{
                        lessThan                (1),
                        lessThanOrEqual         (2),
                        equal                   (3),
                        greaterThanOrEqual      (4),
                        greaterThan             (5),
                        notEqual                (6)},
proximityUnitCode               [5] CHOICE{
                                known   [1] IMPLICIT KnownProximityUnit,
                                private [2] IMPLICIT INTEGER}}
KnownProximityUnit ::= INTEGER{
        character       (1),
        word            (2),
        sentence        (3),
        paragraph       (4),
        section         (5),
        chapter         (6),
        document        (7),
        element         (8),
        subelement      (9),
        elementType     (10),
        byte            (11)    -- Version 3 only
        }
```

```
--End definitions for RPN Query


SearchResponse ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        resultCount                 [23]        IMPLICIT INTEGER,
        numberOfRecordsReturned     [24]        IMPLICIT INTEGER,
        nextResultSetPosition       [25]        IMPLICIT INTEGER,
        searchStatus                [22]        IMPLICIT BOOLEAN,
        resultSetStatus             [26]        IMPLICIT INTEGER{
                                                    subset  (1),
                                                    interim (2),
                                                    none    (3)} OPTIONAL,
        presentStatus                           PresentStatus OPTIONAL,
        records                                 Records OPTIONAL,
-- Following two parameters may be used only if version 3 is in force.
        additionalSearchInfo    [203]           IMPLICIT OtherInformation OPTIONAL,
        otherInfo                               OtherInformation OPTIONAL}

-- Retrieval APDUs
PresentRequest ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        resultSetId                             ResultSetId,
        resultSetStartPoint         [30]        IMPLICIT INTEGER,
        numberOfRecordsRequested    [29]        IMPLICIT INTEGER,
                additionalRanges    [212]       IMPLICIT SEQUENCE OF Range OPTIONAL,
-- additionalRanges may be included only if version 3 is in force.
        recordComposition           CHOICE{
                                    simple      [19]    ElementSetNames,
                                        -- Must choose 'simple' if version 2 is in force
                                    complex     [209]   IMPLICIT CompSpec} OPTIONAL,
        preferredRecordSyntax       [104]       IMPLICIT OBJECT IDENTIFIER OPTIONAL,
        maxSegmentCount             [204]       IMPLICIT INTEGER OPTIONAL,--level 1 or 2
                maxRecordSize       [206]       IMPLICIT INTEGER OPTIONAL,--level 2 only
                maxSegmentSize      [207]       IMPLICIT INTEGER OPTIONAL,--level 2 only
        otherInfo                               OtherInformation OPTIONAL}
Segment ::= SEQUENCE{
-- Segment APDU may only be used when version 3 is in force, and only when segmentation is in effect
        referenceId                             ReferenceId OPTIONAL,
        numberOfRecordsReturned     [24]        IMPLICIT INTEGER,
        segmentRecords              [0]         IMPLICIT SEQUENCE OF NamePlusRecord,
        otherInfo                               OtherInformation OPTIONAL}
PresentResponse ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        numberOfRecordsReturned     [24]        IMPLICIT INTEGER,
        nextResultSetPosition       [25]        IMPLICIT INTEGER,
        presentStatus                           PresentStatus,
        records                                 Records OPTIONAL,
        otherInfo                               OtherInformation OPTIONAL}

--Begin auxiliary definitions for Search and Present APDUs
--Begin definition of records
Records ::= CHOICE{
        responseRecords             [28]        IMPLICIT SEQUENCE OF NamePlusRecord,
        nonSurrogateDiagnostic      [130]       IMPLICIT DefaultDiagFormat,
        multipleNonSurDiagnostics   [205]       IMPLICIT SEQUENCE OF DiagRec}
```

```
NamePlusRecord ::= SEQUENCE{
        name                    [0] IMPLICIT DatabaseName OPTIONAL,
        record                  [1] CHOICE{
                        retrievalRecord         [1] EXTERNAL,
                        surrogateDiagnostic     [2] DiagRec,
                --Must select one of the above two, retrievalRecord or surrogateDiagnostic,
                --unless 'level 2 segmentation' is in effect.
                        startingFragment        [3] FragmentSyntax,
                        intermediateFragment    [4] FragmentSyntax,
                        finalFragment           [5] FragmentSyntax}}
        FragmentSyntax ::= CHOICE{
                        externallyTagged                EXTERNAL,
                        notExternallyTagged             OCTET STRING}
DiagRec ::= CHOICE{
                defaultFormat           DefaultDiagFormat,
-- Must choose defaultFormat if version 2 is in effect
                externallyDefined       EXTERNAL}
DefaultDiagFormat::= SEQUENCE{
        DiagnosticSetId         OBJECT IDENTIFIER,
        condition               INTEGER,
        addinfo                 CHOICE{
                        v2Addinfo       VisibleString,          --Version 2
                        v3Addinfo       InternationalString     --Version 3

-- SEE COMMENT 1
        }}
--End definition of records
Range  ::= SEQUENCE{
        startingPosition                [1] IMPLICIT INTEGER,
        numberOfRecords                 [2] IMPLICIT INTEGER}
ElementSetNames ::= CHOICE {
        genericElementSetName   [0] IMPLICIT InternationalString,
        databaseSpecific        [1] IMPLICIT SEQUENCE OF SEQUENCE{
                        dbName          DatabaseName,
                        esn             ElementSetName}}
PresentStatus ::= [27]      IMPLICIT INTEGER{
        success  (0),
        partial-1       (1),
        partial-2       (2),
        partial-3       (3),
        partial-4       (4),
        failure         (5)}

-- Begin definition of composition specification
CompSpec ::= SEQUENCE{
        selectAlternativeSyntax     [1] IMPLICIT BOOLEAN,
--See comment for recordSyntax, below
        generic         [2] IMPLICIT Specification OPTIONAL,
        dbSpecific              [3] IMPLICIT SEQUENCE OF SEQUENCE{
                        db      [1] DatabaseName,
                        spec    [2] IMPLICIT Specification} OPTIONAL,
         -- At least one of generic and dbSpecific must occur,
        --  and both may occur. If both, then for any record no
```

```
        --  in the list of databases within dbSpecific, generic applies
                recordSyntax      [4] IMPLICIT SEQUENCE OF OBJECT IDENTIFIER OPTIONAL
        -- For each record, the server selects the first record syntax in
         -- this list that it can support.  If the list is exhausted, the server
        -- may select an alternative syntax if selectAlternativeSyntax is 'true'.
        }
Specification ::= SEQUENCE{
        schema CHOICE{
                 oid [1] IMPLICIT OBJECT IDENTIFIER,
                 uri [300] IMPLICIT InternationalString
                -- only if option bit  21 has been negotiated
        } OPTIONAL,
        elementSpec             [2] CHOICE{
                        elementSetName  [1] IMPLICIT InternationalString,
                        externalEspec   [2] IMPLICIT EXTERNAL} OPTIONAL}
-- End definition of composition specification
-- End auxiliary definitions for search and response APDUs

-- Delete APDUs
DeleteResultSetRequest ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        deleteFunction          [32]    IMPLICIT INTEGER{
                        list    (0),
                        all     (1)},
                resultSetList                   SEQUENCE OF ResultSetId OPTIONAL,
                otherInfo                       OtherInformation OPTIONAL}
DeleteResultSetResponse ::= SEQUENCE{
                referenceId                     ReferenceId OPTIONAL,
                deleteOperationStatus   [0]     IMPLICIT DeleteSetStatus,
                deleteListStatuses      [1]     IMPLICIT ListStatuses OPTIONAL,
                numberNotDeleted        [34]    IMPLICIT INTEGER OPTIONAL,
                bulkStatuses            [35]    IMPLICIT ListStatuses OPTIONAL,
                deleteMessage           [36]    IMPLICIT InternationalString OPTIONAL,
                otherInfo                       OtherInformation OPTIONAL}
ListStatuses ::= SEQUENCE OF SEQUENCE{
        id              ResultSetId,
        status          DeleteSetStatus}
DeleteSetStatus ::= [33] IMPLICIT INTEGER{
        success                                 (0),
        resultSetDidNotExist                    (1),
        previouslyDeletedByServer               (2),
        systemProblemAtServer                   (3),
        accessNotAllowed                        (4),
        resourceControlAtClient                 (5),
        resourceControlAtServer                 (6),
        bulkDeleteNotSupported                  (7),
        notAllRsltSetsDeletedOnBulkDlte         (8),
        notAllRequestedResultSetsDeleted        (9),
        resultSetInUse                          (10)}

--Access- and Resource-control APDUs
AccessControlRequest ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        securityChallenge                       CHOICE{
                        simpleForm              [37]    IMPLICIT OCTET STRING,
                        externallyDefined       [0]     EXTERNAL},
```

```
        otherInfo                                   OtherInformation OPTIONAL}


AccessControlResponse ::= SEQUENCE{
        referenceId                                 ReferenceId OPTIONAL,
        securityChallengeResponse                   CHOICE{
                simpleForm                          [38]    IMPLICIT OCTET STRING,
                externallyDefined                   [0]     EXTERNAL} OPTIONAL,
                --Optional only in version 3; mandatory in version 2.
                -- If omitted (in version 3) then diagnostic must occur.
            diagnostic                              [223]   DiagRec OPTIONAL,
                --Version 3 only

        otherInfo                                   OtherInformation OPTIONAL}

ResourceControlRequest ::= SEQUENCE{
        referenceId                                 ReferenceId OPTIONAL,
        suspendedFlag                   [39]        IMPLICIT BOOLEAN OPTIONAL,
        resourceReport                  [40]        ResourceReport OPTIONAL,
        partialResultsAvailable         [41]        IMPLICIT INTEGER{
                                                    subset  (1),
                                                    interim (2),
                                                    none    (3)} OPTIONAL,
        responseRequired                [42]        IMPLICIT BOOLEAN,
        triggeredRequestFlag            [43]        IMPLICIT BOOLEAN OPTIONAL,
        otherInfo                                   OtherInformation OPTIONAL}
ResourceControlResponse ::= SEQUENCE{
        referenceId                                 ReferenceId OPTIONAL,
        continueFlag                    [44]        IMPLICIT BOOLEAN,
        resultSetWanted                 [45]        IMPLICIT BOOLEAN OPTIONAL,
        otherInfo                                   OtherInformation OPTIONAL}
TriggerResourceControlRequest ::= SEQUENCE{
        referenceId                                 ReferenceId OPTIONAL,
        requestedAction                 [46]        IMPLICIT INTEGER{
                                                    resourceReport  (1),
                                                    resourceControl (2),
                                                    cancel          (3)},
prefResourceReportFormat        [47]        IMPLICIT ResourceReportId OPTIONAL,
        resultSetWanted         [48]        IMPLICIT BOOLEAN OPTIONAL,
        otherInfo                           OtherInformation OPTIONAL}
ResourceReportRequest ::= SEQUENCE{
        referenceId                                 ReferenceId OPTIONAL,
            opId                                [210]   IMPLICIT ReferenceId OPTIONAL,
            prefResourceReportFormat            [49]    IMPLICIT ResourceReportId OPTIONAL,
        otherInfo                                   OtherInformation OPTIONAL}
ResourceReportResponse ::= SEQUENCE{
        referenceId                                 ReferenceId OPTIONAL,
        resourceReportStatus                [50]    IMPLICIT INTEGER{
                        success  (0),
                        partial  (1),
                        failure-1 (2),
                        failure-2 (3),
                        failure-3 (4),
                        failure-4 (5),
                        failure-5 (6),
                        failure-6 (7)},
```

```
        resourceReport                  [51]    ResourceReport OPTIONAL,
        otherInfo                               OtherInformation OPTIONAL}
ResourceReport  ::=     EXTERNAL
ResourceReportId        ::=     OBJECT IDENTIFIER

--Scan APDUs
ScanRequest ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
                databaseNames           [3]     IMPLICIT SEQUENCE OF DatabaseName,
                attributeSet                    AttributeSetId OPTIONAL,
-- SEE COMMENT 2
        termListAndStartPoint                   AttributesPlusTerm,
                stepSize                [5]     IMPLICIT INTEGER OPTIONAL,
        numberOfTermsRequested          [6]     IMPLICIT INTEGER,
                preferredPositionInResponse [7] IMPLICIT INTEGER OPTIONAL,
                otherInfo                       OtherInformation OPTIONAL}
ScanResponse ::= SEQUENCE{
        referenceId                             ReferenceId OPTIONAL,
        stepSize                        [3]     IMPLICIT INTEGER OPTIONAL,
        scanStatus                      [4]     IMPLICIT INTEGER {
                                                success         (0),
                                                partial-1       (1),
                                                partial-2       (2),
                                                partial-3       (3),
                                                partial-4       (4),
                                                partial-5       (5),
                                                failure         (6) },
        numberOfEntriesReturned         [5]     IMPLICIT INTEGER,
        positionOfTerm                  [6]     IMPLICIT INTEGER OPTIONAL,
                entries                 [7]     IMPLICIT ListEntries  OPTIONAL,
                attributeSet            [8]     IMPLICIT AttributeSetId OPTIONAL,
-- SEE COMMENT 3
        otherInfo                               OtherInformation OPTIONAL}

--Begin auxiliary definitions for Scan
ListEntries ::= SEQUENCE{
                entries                 [1]     IMPLICIT SEQUENCE OF Entry OPTIONAL,
                nonsurrogateDiagnostics [2]     IMPLICIT SEQUENCE OF DiagRec OPTIONAL
--At least one of entries and nonsurrogateDiagnostics must occur
        }
Entry  ::= CHOICE {
        termInfo                [1]     IMPLICIT TermInfo,
        surrogateDiagnostic     [2]     DiagRec}
TermInfo ::= SEQUENCE {
        term                            Term,
                displayTerm             [0]     IMPLICIT InternationalString OPTIONAL,
        -- See comment 11
        suggestedAttributes             AttributeList OPTIONAL,
        alternativeTerm         [4]     IMPLICIT SEQUENCE OF AttributesPlusTerm OPTIONAL,
        globalOccurrences       [2]     IMPLICIT INTEGER OPTIONAL,
        byAttributes            [3]     IMPLICIT OccurrenceByAttributes OPTIONAL,
        otherTermInfo                   OtherInformation OPTIONAL}
OccurrenceByAttributes ::= SEQUENCE OF SEQUENCE{
        attributes              [1]     AttributeList,
        occurrences                     CHOICE{
                                global          [2]     INTEGER,
```

```
                        byDatabase      [3]     IMPLICIT SEQUENCE OF SEQUENCE{
                                                db              DatabaseName,
                                                num     [1]     IMPLICIT INTEGER OPTIONAL,
                        otherDbInfo     OtherInformation OPTIONAL}} OPTIONAL,
        otherOccurInfo  OtherInformation OPTIONAL}
--End auxiliary definitions for Scan

--Sort APDUs
SortRequest  ::= SEQUENCE{
        referenceId                     ReferenceId OPTIONAL,
        inputResultSetNames     [3]     IMPLICIT SEQUENCE OF InternationalString,
        sortedResultSetName     [4]     IMPLICIT InternationalString,
        sortSequence            [5]     IMPLICIT SEQUENCE OF SortKeySpec,
--Separate instance of SortKeySpec  for each sort key
--Order of occurrence is from major to minor
        otherInfo                       OtherInformation OPTIONAL}
SortResponse ::= SEQUENCE{
        referenceId                     ReferenceId OPTIONAL,
        sortStatus              [3]     IMPLICIT INTEGER{
                                        success  (0),
                                        partial-1 (1),
                                        failure  (2)},
        resultSetStatus         [4]     IMPLICIT INTEGER{
                                        empty           (1),
                                        interim         (2),
                                        unchanged       (3),
                                        none            (4)} OPTIONAL,
                diagnostics             [5]     IMPLICIT SEQUENCE OF DiagRec
                OPTIONAL,
        resultCount             [6] IMPLICIT INTEGER OPTIONAL,
                -- Size of the output result set.
                --Server is never obligated to supply this parameter,
                --there is no default value, and the client should
                -- not draw any conclusion by its omission
        otherInfo                       OtherInformation OPTIONAL}

--Begin auxiliary definitions for Sort
SortKeySpec ::= SEQUENCE{
        sortElement                     SortElement,
        sortRelation            [1]     IMPLICIT INTEGER{
                                        ascending               (0),
                                        descending              (1),
                                        ascendingByFrequency    (3),
                                        descendingByfrequency   (4)},
-- SEE COMMENT 4
        caseSensitivity         [2]     IMPLICIT INTEGER{
                                        caseSensitive           (0),
                                        caseInsensitive         (1)},
        missingValueAction      [3]     CHOICE{
                                        abort   [1] IMPLICIT NULL,
                                        null    [2] IMPLICIT NULL,
--Supply a null value for missing value
        missingValueData        [3] IMPLICIT OCTET STRING} OPTIONAL}
SortElement ::=  CHOICE{
        generic                 [1] SortKey,
        datbaseSpecific         [2] IMPLICIT SEQUENCE OF SEQUENCE{
```

```
                databaseName    DatabaseName,
                dbSort          SortKey}}
SortKey ::= CHOICE{
-- See comment 12
        privateSortKey          [0]     IMPLICIT InternationalString,
        elementSpec             [1]     IMPLICIT Specification,
        sortAttributes          [2]     IMPLICIT SEQUENCE{
                id      AttributeSetId,
                list    AttributeList}}
--End auxiliary definitions for sort

--Extended Service APDUs
ExtendedServicesRequest ::= SEQUENCE{
        referenceId                     ReferenceId OPTIONAL,
        function                [3]     IMPLICIT INTEGER {
                                        create  (1),
                                        delete  (2),
                                        modify  (3)},
        packageType             [4]     IMPLICIT OBJECT IDENTIFIER,
                packageName     [5]     IMPLICIT InternationalString OPTIONAL,
                        --PackageName is mandatory for 'modify' or 'delete'; optional for 'create'.
                        --Following four parameters mandatory for 'create'; should be included on
                        --'modify' if being modified; it is not needed on 'delete'.
                userId          [6]     IMPLICIT InternationalString OPTIONAL,
        retentionTime           [7]     IMPLICIT IntUnit OPTIONAL,
        permissions             [8]     IMPLICIT Permissions OPTIONAL,
        description             [9]     IMPLICIT InternationalString OPTIONAL,
        taskSpecificParameters  [10]    IMPLICIT EXTERNAL OPTIONAL,
                --Mandatory for 'create'; included on 'modify' if specific parameters being modified;
                -- not necessary on 'delete'. For the 'EXTERNAL,' use OID of specific
                --ES definition and select CHOICE [1]: 'esRequest'.
        waitAction                      [11]    IMPLICIT INTEGER{
                                                        wait                    (1),
                                                        waitIfPossible          (2),
                                                        dontWait                (3),
                                                        dontReturnPackage       (4)},
        elements                                IMPLICIT ElementSetName OPTIONAL,
        otherInfo                               IMPLICIT OtherInformation OPTIONAL}
ExtendedServicesResponse ::= SEQUENCE{
        referenceId                     ReferenceId OPTIONAL,
        operationStatus         [3]     IMPLICIT INTEGER{
                                                done            (1),
                                                accepted        (2),
                                                failure         (3)},
                diagnostics     [4]     IMPLICIT SEQUENCE OF DiagRec OPTIONAL,
        taskPackage             [5]     IMPLICIT EXTERNAL OPTIONAL,
        -- Use OID: {Z39-50-recordSyntax (106)} and corresponding syntax.
        -- For the EXTERNAL, 'taskSpecific,' within that definition, use OID
        -- of the specific ES, and choose [2], 'taskPackage'.
        otherInfo                               IMPLICIT OtherInformation OPTIONAL}
Permissions ::= SEQUENCE OF SEQUENCE{
        userId                  [1] IMPLICIT InternationalString,
        allowableFunctions      [2] IMPLICIT SEQUENCE OF INTEGER{
                                        delete                  (1),
                                        modifyContents          (2),
                                        modifyPermissions       (3),
```

```
                                present             (4),
                                invoke              (5)}}

Close ::= SEQUENCE{
        referenceId                                 ReferenceId OPTIONAL,
        -- See 3.2.11.1.5
    closeReason                 CloseReason,
    diagnosticInformation   [3] IMPLICIT InternationalString OPTIONAL,
    resourceReportFormat    [4] IMPLICIT ResourceReportId OPTIONAL,
        --For use by client only, and only on Close request
        --Client requests server to include report in response
    resourceReport          [5] ResourceReport OPTIONAL,
        --For use by server only, unilaterally on Close request
        --On Close response may be unilateral or in response to client request
    otherInfo                       OtherInformation OPTIONAL}
CloseReason ::=         [211] IMPLICIT INTEGER{
    finished                (0),
    shutdown                (1),
    systemProblem           (2),
    costLimit               (3),
    resources               (4),
    securityViolation       (5),
    protocolError           (6),
    lackOfActivity          (7),
    responseToPeer          (8),
    unspecified             (9)}
DuplicateDetectionRequest ::= SEQUENCE {
    referenceId                     ReferenceId OPTIONAL,
    inputResultSetIds           [3] IMPLICIT SEQUENCE OF InternationalString,
    outputResultSetName         [4] IMPLICIT InternationalString,
    applicablePortionOfRecord   [5] IMPLICIT EXTERNAL OPTIONAL,
    duplicateDetectionCriteria  [6] IMPLICIT SEQUENCE OF
                                    DuplicateDetectionCriterion OPTIONAL,
    clustering                  [7] IMPLICIT BOOLEAN OPTIONAL,
            --'true' means "clustered".
            --This parameter may be omitted only if retentionCriteria
            -- CHOICE is 'numberOfEntries' and its value is 1
    retentionCriteria           [8] IMPLICIT SEQUENCE OF RetentionCriterion,
    sortCriteria                [9] IMPLICIT SEQUENCE OF SortCriterion OPTIONAL,
    otherInfo                       OtherInformation OPTIONAL}
DuplicateDetectionCriterion ::= CHOICE{
    levelOfMatch            [1] IMPLICIT INTEGER,
        --A percentage; 1-100
    caseSensitive           [2] IMPLICIT NULL,
    punctuationSensitive    [3] IMPLICIT NULL,
    regularExpression       [4] IMPLICIT EXTERNAL,
    rsDuplicates            [5] IMPLICIT NULL
                                --Values 6-100 reserved for future assignment
    }
RetentionCriterion ::= CHOICE{
    numberOfEntries         [1] IMPLICIT INTEGER,
                                --Greater than 0
    percentOfEntries        [2] IMPLICIT INTEGER,
                                --1-100
    duplicatesOnly          [3] IMPLICIT NULL,
                                --Should not be chosen if clustering is 'true'
```

```
        discardRsDuplicates     [4] IMPLICIT NULL
                                        --Values 5-100 reserved for future assignment
        }
SortCriterion ::= CHOICE{
        mostComprehensive               [1] IMPLICIT NULL,
        leastComprehensive              [2] IMPLICIT NULL,
        mostRecent                      [3] IMPLICIT NULL,
        oldest                          [4] IMPLICIT NULL,
        leastCost                       [5] IMPLICIT NULL,
        preferredDatabases              [6] IMPLICIT SEQUENCE OF InternationalString
                                        --Values 7-100 reserved for future assignment
        }
DuplicateDetectionResponse ::= SEQUENCE {
        referenceId             ReferenceId  OPTIONAL,
        status          [3]     IMPLICIT INTEGER{
                                        success  (0),
                                        failure    (1)},
        resultSetCount  [4]     IMPLICIT INTEGER OPTIONAL,
        diagnostics     [5]     IMPLICIT SEQUENCE OF DiagRec OPTIONAL,
        otherInfo               OtherInformation OPTIONAL}

--Global auxiliary definitions
ReferenceId             ::=             [2]     IMPLICIT OCTET STRING
ResultSetId             ::=             [31]    IMPLICIT InternationalString
ElementSetName          ::=             [103]   IMPLICIT InternationalString
DatabaseName            ::=             [105]   IMPLICIT InternationalString
AttributeSetId          ::=                     OBJECT IDENTIFIER

--OtherInformation
--SEE COMMENT 5
OtherInformation  ::= [201] IMPLICIT SEQUENCE OF SEQUENCE{
        category                        [1]     IMPLICIT InfoCategory OPTIONAL,
        information     CHOICE{
                        characterInfo           [2]     IMPLICIT InternationalString,
                        binaryInfo              [3]     IMPLICIT OCTET STRING,
                        externallyDefinedInfo   [4]     IMPLICIT EXTERNAL,
                        oid                     [5]     IMPLICIT OBJECT IDENTIFIER}}
InfoCategory ::= SEQUENCE{
        categoryTypeId  [1]     IMPLICIT OBJECT IDENTIFIER OPTIONAL,
        categoryValue   [2]     IMPLICIT INTEGER}

--Units
--SEE COMMENT 6
IntUnit ::= SEQUENCE{
        value           [1] IMPLICIT INTEGER,
        unitUsed        [2] IMPLICIT Unit}
Unit ::= SEQUENCE{
        unitSystem              [1] InternationalString OPTIONAL,       --e.g. 'SI'
        unitType                [2] StringOrNumeric OPTIONAL,           --e.g. 'mass'
        unit                    [3] StringOrNumeric OPTIONAL,           --e.g. 'kilograms'
        scaleFactor             [4] IMPLICIT INTEGER OPTIONAL           --e.g. 9 means 10**9
        }

--CharacterString
InternationalString ::= GeneralString
--SEE COMMENT 7
```

```
StringOrNumeric ::= CHOICE{
        string     [1] IMPLICIT InternationalString,
        numeric  [2] IMPLICIT INTEGER}
END IR DEFINITIONS
```

Comments

Comments referred to within the text of the ASN.1 are presented here.

Comment 1

A server should always include addInfo, even if it doesn't have any meaningful additional information to provide, including the case where the diagnostic's definition doesn't even specify the nature of the additional information. This is for historical reasons, for compatibility with earlier versions. (A client may, if it wishes, accept a diagnostic where addInfo is omitted, without considering this to be a protocol error.)
When a diagnostic's definition does not specify the nature of the additional information, it is recommended that the server either supply a text string, for example, "no further information available" (and it is further recommended in this case that the client show the message to the user) or supply a well-known value, in lieu of, and to mean "no additional information available". The well-known value would be either "null" or a zero-length string. The first option is recommended in cases where the server knows what language the client prefers, as in the case where a language has been negotiated. When the server does not know what language the client prefers, the second option is recommended; note, however (for the well-known value) a zero-length string poses a problem for some implementations.
When a diagnostic's definition does specify a particular type of additional information - as for example, diagnostic 109: "database unavailable", which specifies addInfo as "database name" - the use of a well-known value is inappropriate, because it cannot be distinguished from a real value (for example, "null" could be a database name). However, in this case the server should usually be capable of providing the specified information. For example, if a server declares "database unavailable" it should be able to say which database.

Comment 2

This is optional because an attribute set id may be provided within termListandStartPoint, which is of type AttributesPlusTerm, which includes AttributeList, which provides for an attribute set id to be included for every attribute type/value pair. The attribute set id may be omitted here only if an attribute set id is attached individually to every attribute pair. No attribute pair should remain unqualified by an attribute set id. If a Scan request is sent where one or more attribute pairs remains unqualified (that is, where the attribute set id parameter is omitted in the Scan request and is also omitted for one or more attribute pairs) the server should treat this condition as an error. The server, at its discretion, may treat this either as a protocol error or may fail the Scan and return a diagnostic. Diagnostic 1051: " Scan: Attribute set id required -- not supplied" has been defined for this condition.

Comment 3

This pertains to attribute lists referenced elsewhere in the ScanResponse. Both in TermInfo, and in OccurrenceByAttributes, where AttributeList or AttributePlusTerm are referenced respectively, both reference attributeElement, which may or may not include an attribute set identifier; attributeSet specifies the attribute set identifier in cases where it is omitted in attributeElement.

Comment 4

"Ascending by frequency" (or "descending by frequency") is distinguished from "ascending" (or "descending") as illustrated in this example. Suppose we do a search on Moby Dick, get six hits and sort by author, ascending (conventionally ascending, i.e. not by frequency), and the resulting order is:
1. Author: Bradbury, Ray
Title: Moby Dick and that Bride of Time
2. Author: Bradbury, Ray
Title: Moby Dick and the Dandelion Wine

3. Author: Herbert, T. Walter
Title: Moby-Dick and Calvinism
4. Author: Melville, Herman
Title: Moby Dick : or, The whale : Paintings byLeRoy Neiman
5. Author: Melville, Herman
Title: Moby-Dick : or, The whale; Commentary by Howard Mumford Jones
6. Author: Melville, Herman, 1819-1891.
Title: Moby Dick : or, The whale; Introd. by A. S. W. Rosenbach.
If we do a "descending by frequency" sort, the order (of authors) would be Melville, Bradbury, Walker; because Melville had the highest frequency as author (3), Bradbury second (2), and Walker third (1). (Note that the ordering of items for a given author, e.g. for the three Melville items, is determined by the server and cannot necessarily be predicted by the client.)

Comment 5
OtherInfo was originally developed with as much flexibility as possible, since the scope of its usage was not clear at the time. It was anticipated that "categories" would be useful, but no known values for categoryValue have yet been adopted or assigned, and none (known) in use, nor are there any categories (known) in use. The category itself is optional, and if used, categoryTypeId is optional. So there may be:
(1) No category at all,
(2) A category without qualification, or
(3) A category with qualification.
(1) likely will be used predominantly. (2) would be used between partners who have a prior agreement. For (3), the intention is that categoryTypeId would identify some registration agent. It is not intended as a classification mechanism.

Comment 6
IntUnit is used when value and unit are supplied together. Unit, alone, is used when just specifying a unit (without a value). For example, IntUnit is used in Term, in an RPNQuery, or it can be the datatype of an element within a retrieval record. Unit (alone) would be used in an element request, when requesting data be returned according to a particular unit.

Comment 7
When version 2 is in force, this collapses to VisibleString. That is, only characters in the visibleString repertoire may be used. (Datatype compatibility with version 2 is not affected, because references are IMPLICIT.) When version 3 is in force, the semantics of the GeneralString content may be altered by negotiation during initialization. If no such negotiation is in effect, then GeneralString semantics are in force.
InternationalString was defined first in Z39.50-1995, and the intent was to depricate VisibleString. There are asn.1 identifiers from the 1992 version of type visibleString, so the 1995 version stipulates that when v2 is in effect (because if v2 is in effect one of the two systems may be based on the 1992 version), those parameters are restricted to the visibleString repertoire. Choice of tag, 26 or 27, is not an issue where identifiers are defined as IMPLICIT. In two cases they are not, and these are treated separately.

Comment 8
For idAuthentication, the type ANY is retained for compatibility with earlier versions. For interoperability, the following is recommended:

```
IdAuthentication [7] CHOICE{
        open      VisibleString,  --  for compatibility with 1992 version
        idPass    SEQUENCE {
        groupId                   [0]     IMPLICIT InternationalString OPTIONAL,
        userId                    [1]     IMPLICIT InternationalString OPTIONAL,
        password                  [2]     IMPLICIT InternationalString OPTIONAL },
anonymous               NULL,
other                   EXTERNAL
--May use access control formats for 'other'. See Appendix  ACC.
```

Comment 9
Values higher than 'version-3' should be ignored. Both the Initialize request and Initialize Response APDUs include a value string corresponding to the supported versions. The highest common version is selected for use. If there are no versions in common, "Result" in the Init Response should indicate "reject."
Note: Versions 1 and 2 are identical. Systems supporting version 2 should indicate support for version 1 as well, for interoperability with systems that indicate support for version 1 only (e.g. ISO 10163-1991 implementations).

Comment 10
The attribute set definition (attributeSet) describes how these are to be used, for example whether there may be multiple values (for 'list') for a given type, whether they are string or numeric or a combination, the allowable values, semantics for each value, values of 'semanticAction', and semantics for each.

Comment 11
Presence of displayTerm means that term is not considered by the server to be suitable for display, and displayTerm should instead be displayed.
'term' is the actual term in the term list; 'displayTerm' is for display purposes only, and is not an actual term in the term list.

Comment 12
PrivateSortKey is used when the client and server have a prior agreement about what the supplied key means. For example the client may supply the string 'author', where that string doesn't denote any particular field in the record as defined by the schema, and it isn't an attribute (no attribute set id) but the server knows (because of prior agreement) that that string, when supplied as a private sort key, and when applied to the records in question, refers to a specific field.

ElementSpec is a retrieval element of the record, as defined perhaps by a schema, and specified by an element specification, such as eSpec-1, or by an element set name. The element specification or name should resolve to a single element; if it resolves to several elements, the sort key is not well-defined.

The purpose of sortAttributes is to specify a search field; this may require multiple attributes. However if several attributes are supplied, they should resolve to a single abstract access point.

ASN1.2 Z39.50 ASN.1 Object Identifier Definitions for Object Classes

```
ANSI-Z39-50-ObjectIdentifier  DEFINITIONS ::=
BEGIN
Z39-50  OBJECT IDENTIFIER ::=
{ iso (1) member-body (2) US (840) ANSI-standard-Z39.50 (10003)}
-- thus {Z39-50} is shorthand for {1 2 840 10003}
Z39-50-APDU             OBJECT IDENTIFIER ::= {Z39-50 2}      -- See OID.3
-- and {Z39-50 2} is shorthand for {1 2 840  10003 2} and so on.
Z39-50-attributeSet     OBJECT IDENTIFIER ::= {Z39-50 3}      -- See Appendix ATR
Z39-50-diagnostic       OBJECT IDENTIFIER ::= {Z39-50 4}      -- See Appendix DIAG
Z39-50-recordSyntax     OBJECT IDENTIFIER ::= {Z39-50 5}      --See Appendix REC
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
Z39-50-negotiation      OBJECT IDENTIFIER ::= {Z39-50 15}
```

```
Z39-50-query                    OBJECT IDENTIFIER ::= {Z39-50 16}
END
```

ASN1.3 Z39.50 ASN.1 Definition for GeneralDiagnosticContainer

```
GeneralDiagnosticContainer
{Z39-50-diagnosticFormat generalDiagnosticContainer (4)} DEFINITIONS ::=
BEGIN
IMPORTS DiagRec, DefaultDiagFormat FROM Z39-50-APDU-2001;
DiagnosticContainer ::=  SEQUENCE OF DiagRec
END
```

ASN1.4 Z39.50 ASN.1 Definition for Explain

```
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-2001;
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}

    -- See Comment 1

TargetInfo ::= SEQUENCE {
        commonInfo                      [0]     IMPLICIT CommonInfo OPTIONAL,
Key elements follow:
        name                            [1]     IMPLICIT InternationalString,
--See comment 2
--Non-key brief elements follow:
        recent-news                     [2]     IMPLICIT HumanString OPTIONAL,
        icon                            [3]     IMPLICIT IconObject OPTIONAL,
--Element set name “brief-1' is defined for use when client wants icon to be omitted;
--otherwise ‘brief-1' is identical to ‘brief’.
        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 server supports.
--  All objects listed in AccessInfo for any individual database should also be listed here.
        CommonAccessInfo            [19]    IMPLICIT AccessInfo OPTIONAL}
DatabaseInfo ::= SEQUENCE {
            --A server 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,
--See comment 3
        nicknames               [3]     IMPLICIT SEQUENCE OF DatabaseName OPTIONAL,
        icon                    [4]     IMPLICIT IconObject OPTIONAL,
-- Element set name “brief-1' is defined for use when client wants icon to be omitted;
--otherwise ‘brief-1' is identical to ‘brief’
        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 server'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 PTIONAL}
--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 server 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
-- See comment 4
        }
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,
        -- see comment 5
        searchCost      [3] IMPLICIT INTEGER {
        -- see comment 6
                                optimized               (0),
                                normal                  (1),
                                expensive               (2),
                                filter                  (3), } OPTIONAL,
        scanable        [4]     IMPLICIT BOOLEAN,
                                --      'true' means this list can be scanned
                                --      see comment 7
        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
-- See comment 8
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 }
-- See comment 9
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 }
                -- See comment 10
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, server doesn't consider these important.
        estNumberTerms          [5]     IMPLICIT INTEGER OPTIONAL,
        sampleTerms             [6]     IMPLICIT SEQUENCE OF Term OPTIONAL}
ElementSetDetails ::= SEQUENCE {
        -- see comment 11
        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 client 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 server 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 {
        -- SEE COMMENT 12
        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 OPTIONAL,
--Changed to OPTIONAL in 2001 version as result of defect report
--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},
        depricated              [1] IMPLICIT SEQUENCE {
                        depricated0     [0] IMPLICIT InternationalString,
                        depricated1     [1] IMPLICIT InternationalString OPTIONAL,
                        depricated2     [2] IMPLICIT InternationalString OPTIONAL,
                        depricated3     [3] IMPLICIT InternationalString},
-- This element depricated in Z39.50-2001
        other                   [2] IMPLICIT SEQUENCE {
                                        type    [0] IMPLICIT InternationalString,
                                        address [1] IMPLICIT InternationalString}}
AccessInfo ::= SEQUENCE {
        -- see comment 13
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{
                                and             (0),
                                or              (1),
                                and-not         (2),
                                prox            (3)}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 OPTIONA  --Subscription charges
        otherCharges            [5] IMPLICIT SEQUENCE OF SEQUENCE{
                        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
```

Comment 1

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 server system:
The server 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 server may omit any items it doesn't support, for example the description of the bib-1 attribute set may omit attributes that the server does not support under any circumstances.
Databases that may be searched together can be listed in the dbCominations element of the TargetInfo record.

Comment 2
There should generally be only a single TargetInfo record for a given Explain database, so as a practical matter in most cases this record may be considered not to have a key. However in some cases a key may be useful, for example to allow the client to ascertain that it in fact connected to the right server, or is accessing the right Explain database. In any case, to search for the (single) TargetInfo record in an Explain database, a query need not include an operand with Use attribute serverName and term = server name. A single operand with Use attribute ExplainCategory and term = 'TargetInfo' is sufficient.

Comment 3
If explainDatabase is 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.

Comment 4
Each occurrence of equivalentAttributes 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.

Comment 5
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.

Comment 6
Values of searchCost are:
Optimized: The attribute (or combination) associated with this list will do fast searches.
`Normal:` The attribute (combination) will work as expected. So there's probably an index for the attribute (combination) or some similar mechanism.
`Expensive:` 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: Can't search with this attribute (combination) alone.

Comment 7
A term list might exist to optimize searching (i.e. the so-called term list is actually an index) even though it is not scanable. For example, a server maintaining an index of social security numbers might not support scanning that index.

Comment 8
The detail records describe relationships among entities supported by the server. 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.
Client-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.

Comment 9
The human-readable description should generally be provided
It is legal for both default elements to be missing, which means that the server will allow the attribute type to be omitted, but isn't saying what it will do
Comment 10
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.

Comment 11
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 server. 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.

Comment 12
A record in the VariantSetInfo category describes a variant set definition, i.e. classes, types, and values, for a specific variant set definition supported by the server. Support by the server of a particular variant set definition does not imply that the definition is supported for any specific database or element.

Comment 13
AccessInfo contains the fundamental information about what facilities are required to use this server or server. For example, if a client can handle none of the record syntaxes a database can provide, it might choose not to access the database.

ASN1.5 Z39.50 ASN.1 Definition for SUTRS

```
RecordSyntax-SUTRS
{Z39-50-recordSyntax sutrs (101)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString  FROM Z39-50-APDU-2001;
SutrsRecord ::=  InternationalString
-- See Comment 1.
END
```

Comment 1
This use of InternationalString should be interpreted to mean GeneralString
Line terminator is ASCII LF (X'0A')
Recommended maximum line length is 72 characters

ASN1.6 Z39.50 ASN.1 Definition for GRS-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-2001;
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, server not telling or it is irrelevant.
-- 1-based. Tags are numbered beginning with 1.
        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,
--Note: VisibleString (ASN.1 tag 26) is not supported by GRS-1.
--The tag for 'string' must always be 2, even when
--version 2 is in effect and the character repertoire collapses to that of VisibleString.
        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 client can use for display
        supportedVariants               [5]     IMPLICIT SEQUENCE OF Variant OPTIONAL,
        message                         [6]     IMPLICIT InternationalString OPTIONAL,
        elementDescriptor               [7]     IMPLICIT OCTET STRING OPTIONAL,
            --For example, a DTD
        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
            -- See comment 1.
```

```
        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,
        serverToken             [4] IMPLICIT OCTET STRING OPTIONAL
            --Client 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
```

Comment 1
surrogateFor and surrogateElement are for use when a record contains a number of elements. For example, suppose the elements are objects and one particular object is a surrogate (e.g. thumbnail) for another element within that record. Suppose for element A surrogateFor is supplied; it is supplied in the form of a tagPath, suppose for element B. This means that element A is a surrogate for element B. Or suppose for element A, surrogateElement is supplied, in the form of a tagPath for element B. This means that element B is a surrogate for element A. (Note that these two are useful only when both the element and its surrogate are in the same record.)

ASN1.7 Z39.50 ASN.1 Definition for the Extended Services Task Package Record Syntax

```
RecordSyntax-ESTaskPackage
{Z39-50-recordSyntax esTaskPackage (106)} DEFINITIONS ::=
BEGIN
IMPORTS Permissions, InternationalString, IntUnit, DiagRec FROM Z39-50-APDU-2001;
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,
        serverReference         [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
```

ASN1.8 Z39.50 ASN.1 Definition for Resource Report Format Resource-2

```
ResourceReport-Format-Resource-2
{Z39-50-resourceReport resource-2 (2)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString, StringOrNumeric, IntUnit FROM Z39-50-APDU-2001;
ResourceReport ::= SEQUENCE{
        estimates       [1]     IMPLICIT SEQUENCE OF Estimate OPTIONAL,
        message         [2]     IMPLICIT InternationalString OPTIONAL}
Estimate ::= SEQUENCE{
        type            [1]     StringOrNumeric,
            --Numeric values of 1-16 are the same as used in Resource-1.
        value           [2]     IMPLICIT IntUnit
```

```
-- When expressing currency: unitSystem (of Unit) is 'z3950'
-- (case insensitive), and unitType is 'iso4217-1990' (case insensitive).
-- Unit is currency code from ISO 4217-1990
        }
        END
```

ASN1.9 Z39.50 ASN.1 Definition for Access Control Formats

ASN1.9.1 Prompt-1

```
AccessControlFormat-prompt-1
{Z39-50-accessControl prompt-1 (1)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString, DiagRec FROM Z39-50-APDU-2001;
PromptObject ::= CHOICE{
        challenge               [1] IMPLICIT Challenge,
        response                [2] IMPLICIT Response}
Challenge ::= SEQUENCE OF SEQUENCE {
        promptId                [1] PromptId,
            -- See comment 1
        defaultResponse         [2] IMPLICIT InternationalString OPTIONAL,
        promptInfo              [3] CHOICE{
        character               [1] IMPLICIT InternationalString,
        encrypted               [2] IMPLICIT Encryption} OPTIONAL,

                -- See comment 2
        regExpr                 [4] IMPLICIT InternationalString OPTIONAL,
                -- See comment 3
        responseRequired        [5] IMPLICIT NULL OPTIONAL,
        allowedValues           [6] IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
            --e.g. promptId="Desired color"; allowed = 'red', 'blue','Green'
        shouldSave                      [7] IMPLICIT NULL OPTIONAL,
                -- See comment 4
        dataType                [8] IMPLICIT INTEGER{
                                        integer         (1),
                                        date            (2),
                                        float           (3),
                                        alphaNumeric    (4),
                                        url-urn         (5),
                                        boolean         (6)} OPTIONAL,
                -- See comment  5
        diagnostic                      [9] IMPLICIT EXTERNAL OPTIONAL
                --Intended for repeat requests when there is an error
                --the client should report to the user from previous attempt.
        }
Response ::= SEQUENCE OF SEQUENCE {
        promptId                [1] PromptId,
                -- See comment 6
        promptResponse          [2] CHOICE{
        string                  [1] IMPLICIT InternationalString,
        accept                  [2] IMPLICIT BOOLEAN,
        acknowledge             [3] IMPLICIT NULL,
        diagnostic              [4] DiagRec,
        encrypted               [5] IMPLICIT Encryption}}
```

```
PromptId ::= CHOICE{
        enummeratedPrompt               [1] IMPLICIT SEQUENCE{
                        type                    [1] IMPLICIT INTEGER{
                                                        groupId         (0),
                                                        userId          (1),
                                                        password        (2),
                                                        newPassword     (3),
                                                        copyright       (4),
                                                                -- See comment 7
                                                        sessionId       (5)},
                        suggestedString         [2] IMPLICIT InternationalString OPTIONAL},
        nonEnumeratedPrompt     [2] IMPLICIT InternationalString}
Encryption ::= SEQUENCE{
        cryptType       [1] IMPLICIT OCTET STRING OPTIONAL,
        credential      [2] IMPLICIT OCTET STRING OPTIONAL,
            --Random number, SALT, or other factor
        data            [3] IMPLICIT OCTET STRING}
        END
```

Comment 1
Server supplies a number (for an enumerated prompt) or string (for a non-enumerated prompt), for each prompt, and the client returns it in response, for this prompt, so server may correlate the prompt response with the prompt.

Comment 2
Information corresponding to an enumerated prompt. For example if 'type', within PromptId, is 'copyright', then promptInfo may contain a copyright statement

Comment 3
A regular expression that promptResponse should match. See IEEE 1003.2 Volume 1, Section 2.8 "Regular Expression Notation." For example if promptId is "Year of publication," regExpr might be "19[89][0-9]|20[0-9][0-9]".

Comment 4
Server recommends that client save the data that it prompts from the user corresponding to this prompt, because it is likely to be requested again (so client might not have to prompt the user next time).

Comment 5
Server telling client type of data it wants. E.g., if "date" is specified, presumably the client will try to prompt something "date-like" from the user.

Comment 6
Corresponds to a prompt in the challenge, or may be unprompted. For example "newPassword," if unprompted, should be "enumerated." If this responds to a non-enumerated prompt, then nonEnumeratedPrompt should contain the prompt string from the challenge.

Comment 7
When type on Challenge is 'copyright', promptInfo has text of copyright message to be displayed verbatim to the user. If promptResponse indicates 'acceptance', this indicates the user has been shown, and accepted, the terms of the copyright. This is not intended to be legally binding, but provides a good-faith attempt on the part of the server to inform the user of the copyright.

ASN1.9.2 Des-1

```
AccessControlFormat-des-1
{Z39-50-accessControlFormat  des-1 (2)} DEFINITIONS ::=
BEGIN
DES-RN-Object ::= CHOICE {
        challenge               [1] IMPLICIT DRNType,
        response                [2] IMPLICIT DRNType}
DRNType ::= SEQUENCE{
        userId                  [1] IMPLICIT OCTET STRING OPTIONAL,
        salt                    [2] IMPLICIT OCTET STRING OPTIONAL,
        randomNumber            [3] IMPLICIT OCTET STRING}
        END
```

ASN1.9.3 Krb-1

```
AccessControlFormat-krb-1
{Z39-50-accessControlFormat  krb-1 (3)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString FROM Z39-50-APDU-2001;
KRBObject ::= CHOICE {
        challenge       [1] IMPLICIT KRBRequest,
        response        [2] IMPLICIT KRBResponse}
KRBRequest ::= SEQUENCE{
        service         [1] IMPLICIT InternationalString,
        instance        [2] IMPLICIT InternationalString OPTIONAL,
        realm           [3] IMPLICIT InternationalString OPTIONAL}
            --Server requests a ticket for the given service, instance, and realm
KRBResponse ::= SEQUENCE{
        userid          [1] IMPLICIT InternationalString OPTIONAL,
        ticket          [2] IMPLICIT OCTET STRING}
            -- Client responds with a ticket for the requested service
END
```

ASN1.10 Z39.50 ASN.1 Definitions for Extended Services

```
ESFormat-PersistentResultSet
{Z39-50-extendedServic e persistentResultSet (1)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString FROM Z39-50-APDU-2001;
PersistentResultSet ::= CHOICE{
        esRequest       [1] IMPLICIT SEQUENCE{
                                toKeep          [1] IMPLICIT NULL,
                                notToKeep       [2] ClientPartNotToKeep OPTIONAL},
        taskPackage     [2] IMPLICIT SEQUENCE{
                                clientPart      [1] IMPLICIT NULL,
                                serverPart      [2] ServerPart OPTIONAL}}
ClientPartNotToKeep ::= SEQUENCE{
        clientSuppliedResultSet     [1] IMPLICIT InternationalString OPTIONAL,
            --Name of transient result set, supplied on request,
```

```
            -- mandatory unless function is 'delete'
        replaceOrAppend         [2] IMPLICIT INTEGER{ -- Only if function is "modify"
                                replace         (1),
                                append          (2)} OPTIONAL}
ServerPart ::= SEQUENCE{
        serverSuppliedResultSet [1] IMPLICIT InternationalString OPTIONAL,
                --Name of transient result set, supplied by server,
                 --representing the persistent result set to which
                --package pertains. Meaningful only when package is
                --presented. (i.e. not on ES response).
        NumberOfRecords         [2] IMPLICIT INTEGER OPTIONAL}
        END


ESFormat-PersistentQuery
{Z39-50-extendedService persistentQuery (2)} DEFINITIONS ::=
BEGIN
IMPORTS Query, InternationalString, OtherInformation FROM Z39-50-APDU-2001;
PersistentQuery ::= CHOICE{
        esRequest       [1] IMPLICIT SEQUENCE{
                        toKeep          [1] ClientPartToKeep OPTIONAL,
                        notToKeep       [2] ClientPartNotToKeep},
        taskPackage     [2] IMPLICIT SEQUENCE{
                        clientPart      [1] ClientPartToKeep OPTIONAL,
                        serverPart      [2] ServerPart}}
ClientPartToKeep ::= SEQUENCE{
                dbNames         [2] IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
        additionalSearchInfo    [3] OtherInformation OPTIONAL}
ClientPartNotToKeep ::= CHOICE{
                package         [1] IMPLICIT InternationalString,
                query           [2] Query}
ServerPart ::= Query
END


ESFormat-PeriodicQuerySchedule
{Z39-50-extendedService periodicQuerySchedule (3)} DEFINITIONS ::=
BEGIN
IMPORTS Query, InternationalString, IntUnit FROM Z39-50-APDU-2001
ExportSpecification, Destination FROM ESFormat-ExportSpecification;
PeriodicQuerySchedule ::= CHOICE{
        esRequest       [1] IMPLICIT SEQUENCE{
                        toKeep          [1] ClientPartToKeep,
                        notToKeep       [2] ClientPartNotToKeep},
        taskPackage     [2] IMPLICIT SEQUENCE{
                        clientPart      [1] ClientPartToKeep,
                        serverPart      [2] ServerPart}}
ClientPartToKeep ::=SEQUENCE{
        activeFlag              [1] IMPLICIT BOOLEAN,
        databaseNames           [2] IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
                --databaseNames must not occur if option bit 20 is set
        resultSetDisposition    [3] IMPLICIT INTEGER{
                                replace         (1),
                                append          (2),
                                createNew       (3)
                                --Only if client and server have agreement about naming
```

```
                        --convention for the resulting package,
                        --and only if no result set is specified
                    } OPTIONAL,
                    --Mandatory on 'create' if result set is specified,
                    --in which case it must be 'replace' or 'append
        alertDestination        [4] Destination OPTIONAL,
        exportParameters        [5] CHOICE{
                                    packageName     [1]  IMPLICIT InternationalString,
                                    exportPackage   [2] ExportSpecification} OPTIONAL}
ClientPartNotToKeep ::= SEQUENCE{
            DatabaseNames  [0] IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
                --Must not occur unless option bit 20 is set
        querySpec                   [1] CHOICE{
                    actualQuery     [1] Query,
                    packageName     [2] IMPLICIT InternationalString} OPTIONAL,
                        --Mandatory for 'create'
            clientSuggestedPeriod   [2] Period OPTIONAL,
                        -- mandatory for 'create'
            expiration              [3] IMPLICIT GeneralizedTime OPTIONAL,
            resultSetPackage        [4] IMPLICIT InternationalString OPTIONAL,
            additionalSearchInfo    [5] OtherInformation OPTIONAL
                --Must not occur unless option bit 20 is set
        }
ServerPart ::= SEQUENCE{
            DatabaseNames  [0] IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
                --databaseNames must occur if bit 20 is set
                -- and must not occur if option bit 20 is not set
        actualQuery             [1] Query,
        serverStatedPeriod      [2] Period,
            --Server supplies the period, which may be same as client proposed
        expiration              [3] IMPLICIT GeneralizedTime OPTIONAL,
            --Server supplies value for task package.
            --It may be the same as client proposed or different from
            --(and overrides) client proposal, but if omitted, there is no expiration.
        resultSetPackage        [4] IMPLICIT InternationalString OPTIONAL,
            --May be omitted only if exportParameters was supplied.
            --Server supplies same name as client supplied, if client did supply a name.
        lastQueryTime           [5] IMPLICIT GeneralizedTime OPTIONAL,
        lastResultNumber        [6] IMPLICIT INTEGER OPTIONAL,
            --Above two were made optional in 2001 version,
            --because there won't be any value for these between
            --the time the package is created and the first query is executed.
        numberSinceModify       [7] IMPLICIT INTEGER OPTIONAL,
        additionalSearchInfo    [8] OtherInformation OPTIONAL
            --Must not occur unless option bit 20 is set
        }
Period ::= CHOICE{
        unit                    [1] IMPLICIT IntUnit,
        businessDaily           [2] IMPLICIT NULL,
        continuous              [3] IMPLICIT NULL,
        other                   [4] IMPLICIT InternationalString}
        END


ESFormat-ItemOrder
{Z39-50-extendedService itemOrder (4)} DEFINITIONS ::=
```

```
BEGIN
IMPORTS InternationalString FROM Z39-50-APDU-2001;
ItemOrder ::= CHOICE{
        EsRequest       [1] IMPLICIT SEQUENCE{
                        toKeep  [1] ClientPartToKeep OPTIONAL,
                        notToKeep       [2] ClientPartNotToKeep},
        taskPackage     [2] IMPLICIT SEQUENCE{
                        clientPart      [1] ClientPartToKeep OPTIONAL,
                        serverPart      [2] ServerPart}}
ClientPartToKeep ::= SEQUENCE{
        supplDescription  [1] IMPLICIT EXTERNAL OPTIONAL,
        contact         [2] IMPLICIT SEQUENCE{
                                name    [1] IMPLICIT InternationalString OPTIONAL,
                                phone   [2] IMPLICIT InternationalString OPTIONAL,
                                email   [3] IMPLICIT InternationalString OPTIONAL} OPTIONAL,
        addlBilling     [3] IMPLICIT SEQUENCE{
                        paymentMethod           [1]   CHOICE{
                                billInvoice                     [0] IMPLICIT NULL,
                                prepay                          [1] IMPLICIT NULL,
                                depositAccount                  [2] IMPLICIT NULL,
                                creditCard                      [3] IMPLICIT CreditCardInfo,
                                cardInfoPreviouslySupplied      [4] IMPLICIT NULL,
                                privateKnown                    [5] IMPLICIT NULL,
                                privateNotKnown                 [6] IMPLICIT EXTERNAL},
                        customerReference       [2] IMPLICIT InternationalString OPTIONAL,
                                --An identifier assigned by the client
                                --to identify the customer.
                                --It could be used when the client want
                                --to search for Item Order task packages
                                --for a specific customer.
                        customerPONumber        [3] IMPLICIT InternationalString OPTIONAL
                        --A purchase order number assigned by the
                        --customer (as opposed to one that might be
                        --assigned by the supplier). Similarly, a client
                        --may search for a task package knowing
                        --only the customer reference.
                    }
                    OPTIONAL}
CreditCardInfo    ::= SEQUENCE{
        nameOnCard              [1] IMPLICIT InternationalString,
        expirationDate          [2] IMPLICIT InternationalString,
        cardNumber              [3] IMPLICIT InternationalString}
ClientPartNotToKeep ::= SEQUENCE{
--Corresponds to 'requestedItem' in service definition
--Must supply at least one, and may supply both.
        resultSetItem   [1] IMPLICIT SEQUENCE{
                        resultSetId     [1] IMPLICIT InternationalString,
                        item            [2] IMPLICIT INTEGER} OPTIONAL,
        itemRequest     [2] IMPLICIT EXTERNAL OPTIONAL
        -- See comment  1.
        }
ServerPart ::= SEQUENCE{
        itemRequest             [1]     IMPLICIT EXTERNAL OPTIONAL,
            --When itemRequest is an ILL-Request APDU,
            --use OID 1.0.10161.2.1 (as above)
        statusOrErrorReport     [2] IMPLICIT EXTERNAL OPTIONAL
```

```
        --When statusOrErrorReport is an ILL Status-Or-Error-Report
        --APDU, use OID 1.0.10161.2.1 (as above)
    auxiliaryStatus         [3] IMPLICIT INTEGER{
                                notReceived         (1),
                                loanQueue           (2),
                                forwarded           (3),
                                unfilledCopyright   (4),
                                filledCopyright     (5)} OPTIONAL}
    END
```

Comment 1

When itemRequest is an ILL-Request APDU, use OID {iso standard 10161 abstract-syntax (2) ill-APDUs (1)}. Note that the 'itemRequest', as EXTERNAL, is not constrained (by Z39.50) to be ASN.1 BER. If the itemRequest is an ILL APDU (and the external uses OID 1.0.10161.2.1) then it is constrained to be ASN.1, but not constrained to be BER; it can be EDIFACT, as provided by the ILL standard. If BER is used, the BER encoding may be wrapped in Base64 (in store and forward mode).

```
ESFormat-Update
{Z39-50-extendedService update (5)} DEFINITIONS ::=
BEGIN
IMPORTS DiagRec, InternationalString FROM Z39-50-APDU-2001;
Update ::= CHOICE{
    esRequest       [1] IMPLICIT SEQUENCE{
                    toKeep  [1] ClientPartToKeep,
                    notToKeep       [2] ClientPartNotToKeep},
    taskPackage     [2] IMPLICIT SEQUENCE{
                    clientPart      [1] ClientPartToKeep,
                    serverPart      [2] ServerPart}}
ClientPartToKeep ::= SEQUENCE{
    action          [1] IMPLICIT INTEGER{
                                recordInsert    (1),
                                recordReplace   (2),
                                recordDelete    (3),
                                elementUpdate   (4)},
    databaseName    [2] IMPLICIT InternationalString,
    schema          [3] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
    elementSetName  [4] IMPLICIT InternationalString OPTIONAL}
ClientPartNotToKeep ::= SuppliedRecords
ServerPart ::= SEQUENCE{
    updateStatus            [1] IMPLICIT INTEGER{
                                    success  (1),
                                    partial  (2),
                                    failure  (3)},
    globalDiagnostics       [2] IMPLICIT SEQUENCE OF DiagRec OPTIONAL,
            --These are non-surrogate diagnostics relating
            --to the task, not to individual records.
            taskPackageRecords      [3] IMPLICIT SEQUENCE OF TaskPackageRecordStructure
    --See comment 1.
    }
--Auxiliary definitions for Update
SuppliedRecords ::= SEQUENCE OF SEQUENCE{
    recordId        [1] CHOICE{
                            number  [1] IMPLICIT INTEGER,
                            string  [2] IMPLICIT InternationalString,
```

```
                        opaque   [3] IMPLICIT OCTET STRING} OPTIONAL,
supplementalId          [2] CHOICE{
                        timeStamp        [1] IMPLICIT GeneralizedTime,
                        versionNumber    [2] IMPLICIT InternationalString,
                        previousVersion  [3] IMPLICIT EXTERNAL} OPTIONAL,
correlationInfo         [3] IMPLICIT CorrelationInfo OPTIONAL,
record                  [4] IMPLICIT EXTERNAL}
CorrelationInfo ::= SEQUENCE{
--Client may supply one or both for any record:
        note    [1] IMPLICIT InternationalString OPTIONAL,
        id      [2] IMPLICIT INTEGER OPTIONAL}
TaskPackageRecordStructure ::= SEQUENCE{
        recordOrSurDiag         [1] CHOICE {
                                record  [1] IMPLICIT EXTERNAL,
--Choose 'record' if recordStatus is 'success', and elementSetName was supplied.
                        diagnostic      [2] DiagRec
-- Choose 'diagnostic', if RecordStatus is failure
                    } OPTIONAL,
        --See comment 2
        correlationInfo         [2] IMPLICIT CorrelationInfo OPTIONAL,
-- This should be included if it was supplied by the client
        recordStatus            [3] IMPLICIT INTEGER{
                                success         (1),
                                queued          2),
                                inProcess       (3),
                                failure         (4)}}
        END
```

Comment 1
There should be a TaskPackageRecordStructure for every record supplied. The server should create such a structure for every record immediately upon creating the task package to include correlation information and status. The record itself would not be included until processing for that record is complete.

Comment 2
The parameter recordOrSurDiag will thus be omitted only if 'elementSetName' was omitted and recordStatus is 'success'; or if record status is 'queued' or in 'process'

```
ESFormat-ExportSpecification
{Z39-50-extendedService exportSpecification (6)} DEFINITIONS ::=
BEGIN
EXPORTS ExportSpecification, Destination; IMPORTS CompSpec, InternationalString FROM
Z39-50-APDU-2001;
ExportSpecification ::= CHOICE{
        esRequest       [1] IMPLICIT SEQUENCE{
                        toKeep          [1] ClientPartToKeep,
                        notToKeep       [2] IMPLICIT NULL},
        taskPackage     [2] IMPLICIT SEQUENCE{
                        clientPart      [1] ClientPartToKeep,
                        serverPart      [2] IMPLICIT NULL}}
ClientPartToKeep ::= SEQUENCE{
        composition             [1] IMPLICIT CompSpec,
        exportDestination       [2] Destination}
Destination ::= CHOICE{
        phoneNumber             [1]   IMPLICIT InternationalString,
        faxNumber               [2]   IMPLICIT InternationalString,
```

```
        x400address             [3]  IMPLICIT InternationalString,
        emailAddress            [4]  IMPLICIT InternationalString,
        pagerNumber             [5]  IMPLICIT InternationalString,
        ftpAddress              [6]  IMPLICIT InternationalString,
        ftamAddress             [7]  IMPLICIT InternationalString,
        printerAddress          [8]  IMPLICIT InternationalString,
        other                   [100]  IMPLICIT SEQUENCE{
                                vehicle      [1] IMPLICIT InternationalString  OPTIONAL,
                                destination  [2] IMPLICIT InternationalString}}
END


ESFormat-ExportInvocation
{Z39-50-extendedService exportInvocation (7)} DEFINITIONS ::=
BEGIN
IMPORTS InternationalString, IntUnit FROM Z39-50-APDU-2001
ExportSpecification FROM ESFormat-ExportSpecification;
ExportInvocation ::= CHOICE{
        esRequest       [1] IMPLICIT SEQUENCE{
                        toKeep          [1] ClientPartToKeep,
                        notToKeep       [2] ClientPartNotToKeep},
        taskPackage     [2] IMPLICIT SEQUENCE{
                        clientPart      [1] ClientPartToKeep,
                        serverPart      [2] ServerPart OPTIONAL}}
ClientPartToKeep ::= SEQUENCE{
        exportSpec      [1] CHOICE{
                        packageName     [1] IMPLICIT InternationalString,
                        packageSpec     [2] ExportSpecification},
        numberOfCopies  [2] IMPLICIT INTEGER}
ClientPartNotToKeep     ::= SEQUENCE{
        resultSetId             [1] IMPLICIT InternationalString,
        records                 [2] CHOICE{
                                all     [1] IMPLICIT NULL,
                                ranges  [2] IMPLICIT SEQUENCE OF SEQUENCE{
                                        start   [1] IMPLICIT INTEGER,
                                        count   [2] IMPLICIT INTEGER OPTIONAL
                                --Count may be omitted only on last range,
                                -- to indicate "all remaining records beginning with 'start'."
                            }}}
ServerPart      ::= SEQUENCE{
        estimatedQuantity       [1] IMPLICIT IntUnit OPTIONAL,
        quantitySoFar           [2] IMPLICIT IntUnit OPTIONAL,
        estimatedCost           [3] IMPLICIT IntUnit OPTIONAL,
        costSoFar               [4] IMPLICIT IntUnit OPTIONAL}
END
```

ASN1.11. Z39.50 ASN.1 Definition for SearchResult-1

```
UserInfoFormat-searchResult-1
{Z39-50-userInfoFormat searchResult-1 (1)} DEFINITIONS ::=
BEGIN
IMPORTS DatabaseName, Term, Query, IntUnit, InternationalString FROM Z39-50-APDU-2001;
SearchInfoReport ::= SEQUENCE OF SEQUENCE{
```

```
            subqueryId          [1] IMPLICIT InternationalString OPTIONAL,
                    --Shorthand identifier of subquery
            fullQuery           [2] IMPLICIT BOOLEAN,
                    --'true' means this is the full query; 'false', a sub-query
        subqueryExpression      [3] QueryExpression OPTIONAL,
            --A subquery of the query as submitted.
            --May be whole query; if so, "fullQuery" should be 'true'
        subqueryInterpretation  [4] QueryExpression OPTIONAL,
            --How server interpreted subquery
        subqueryRecommendation  [5] QueryExpression OPTIONAL,
            -- Server-recommended alternative
        subqueryCount           [6] IMPLICIT INTEGER OPTIONAL,
            --Number of records for this subQuery, across
            --all of the specified databases. (If during search,
            --via resource control, number of records so far)
        subqueryWeight          [7] IMPLICIT IntUnit OPTIONAL,
            --Relative weight of this subquery
        resultsByDB             [8] IMPLICIT ResultsByDB OPTIONAL}

        ResultsByDB ::= SEQUENCE OF SEQUENCE{
            databases           [1] CHOICE{
                                    all         [1] IMPLICIT NULL,
                                    --Applies across all of the databases in Search APDU
                                    list                [2] IMPLICIT SEQUENCE OF DatabaseName
                                    --Applies across all databases in this list
                                    },
            count               [2] IMPLICIT INTEGER OPTIONAL,
                    --Number of records for query component
                    --(and, as above, if during search, via resource control,
                    -- number of records so far)
            resultSetName           [3] IMPLICIT InternationalString OPTIONAL
            --See comment 1.
            }
        QueryExpression ::=  CHOICE {
            term    [1] IMPLICIT SEQUENCE{
                    queryTerm       [1]     Term,
                    termComment     [2] IMPLICIT InternationalString OPTIONAL},
            query   [2] Query}
END
```

Comment 1
Server-assigned result set by which subQuery is available. Should not be provided unless processing for this query component is concluded (i.e., when this report comes during search, via resource control, as opposed to after search, via additionalSearchInfo)

ASN1.12. Z39.50 ASN.1 Definition for UserInfo-1

```
UserInfoFormat-userInfo-1
{Z39-50-userInfoFormat userInfo-1 (3)} DEFINITIONS ::=
BEGIN
IMPORTS OtherInformation FROM Z39-50-APDU-2001;
UserInfo-1 ::= OtherInformation
END
```

ASN1.13. Z39.50 ASN.1 Definitions for eSpec-2

```
{Z39-50-elementSpec eSpec-2 (2)} DEFINITIONS ::=
--For detailed semantics, see Appendix RET.
BEGIN
IMPORTS Variant FROM RecordSyntax-generic
StringOrNumeric, InternationalString FROM Z39-50-APDU-2001;
Espec-2 ::= SEQUENCE{
elementSetNames          [1] IMPLICIT SEQUENCE OF InternationalString OPTIONAL,
        -- See comment 1
        defaultVariantSetId         [2] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
                --If supplied, applies whenever variantRequest does not include variantSetId
        defaultVariantRequest       [3] IMPLICIT Variant OPTIONAL,
                --See comment 2.
        defaultTagType              [4] IMPLICIT INTEGER OPTIONAL,
                 --If supplied, applies whenever 'tagType'
                 --(within 'tag' within TagPath) is omitted
        elements                    [5] IMPLICIT SEQUENCE OF ElementRequest OPTIONAL}
ElementRequest::= CHOICE{
        SimpleElement               [1] IMPLICIT SimpleElement,
        compositeElement            [2] IMPLICIT SEQUENCE{
                                        elementList         [1] CHOICE{
                                                primitives          [1] IMPLICIT
                                                SEQUENCE OF InternationalString,
                        --Client may specify one or more element set names,
                        --each identifying a set of elements, and the composite element is the union
                                                specs               [2] IMPLICIT
                                                SEQUENCE OF SimpleElement},
                                        deliveryTag         [2] IMPLICIT TagPath,
                                                --DeliveryTag tagPath for compositeElement
                                                --may not include wildThing or wildPath
                                        VariantRequest      [3] IMPLICIT Variant OPTIONAL}}
SimpleElement ::= SEQUENCE{
        path                        [1] IMPLICIT TagPath,
        variantRequest              [2] IMPLICIT Variant OPTIONAL}
TagPath ::= SEQUENCE OF CHOICE{
        specificTag                 [1] IMPLICIT SEQUENCE{
                                        --The following line, schemaId is the
                                        --only difference in this definition from that of eSpec-1.
                        schemaId            [0] IMPLICIT OBJECT  IDENTIFIER OPTIONAL,
                                        --see comment 3
                        TagType             [1] IMPLICIT INTEGER OPTIONAL,
                        --If omitted, then 'defaultTagType' (above) applies,
                        --if supplied, and if not supplied, then default
                        --listed in schema applies
                        TagValue            [2] StringOrNumeric,
                        occurrence          [3] Occurrences OPTIONAL
                                        --default is "first occurrence"
                },
        wildThing                   [2] Occurrences,
        -- See comment 4
        wildPath                    [3] IMPLICIT NULL
        -- See comment 5.
Occurrences ::= CHOICE{
        all                         [1] IMPLICIT NULL,
```

```
        last                [2] IMPLICIT NULL,
        values              [3] IMPLICIT SEQUENCE{
                            start       [1] IMPLICIT INTEGER,
                            --If 'start' alone is included, then
                            --single occurrence is requested
                    howMany         [2] IMPLICIT INTEGER OPTIONAL
                    --For example, if 'start' is 5 and 'howMany' is 6,
                    --then request is for "occurrences 5 through 10."
                    }}
END
```

Comment 1
Client may include one or more element set names, each specifying a set of elements. Each of the elements is to be treated as an elementRequest in the form of simpleElement, where occurrence is 1.

Comment 2
If defaultVariantRequest is supplied, then for each simple elementRequest that does not include a variantRequest, the defaultVariantRequest applies. (defaultVariantRequest does not apply to a compositeRequest.)

Comment 3
SchemaId occurs only if the tagType in specificTag is to be interpreted according to some schema other than that which was specified in CompSpec (which is what references eSpec). The optional schema id is attached at the tag level. Its purpose is to qualify the tagType only.

Comment 4
Wildthing: Get Nth "thing" at this level, regardless of tag, for each N specified by "Occurrences" (which may be 'all' meaning match every element at this level). E.g., if "Occurrences" is 3, get third element regardless of its tag or the tag of the first two elements.

Comment 5
Wildpath: Match any tag, at this level or below, that is on a path for which next tag in this TagPath sequence occurs. WildPath may not be last member of the TagPath sequence.

ASN1.14. Z39.50 ASN.1 Definitions for eSpec-q

```
Element Specification eSpec-q
{Z39-50-elementSpec eSpec-q (3)} DEFINITIONS ::=
BEGIN
IMPORTS Term,  AttributeList, AttributeElement
FROM Z39-50-APDU-2001
Espec-q ::= SEQUENCE{
        valueRestrictor         [1] IMPLICIT ValueRestrictor,
        elementSelector         [2] IMPLICIT EXTERNAL OPTIONAL}
ValueRestrictor ::= SEQUENCE{
        attributeSetId          OBJECT IDENTIFIER,
        nodeSelectionCriteria   RPNStructure}
RPNStructure ::= CHOICE{
        op                              [0] AttributesPlusTerm,
        rpnRpnOp                        [1] IMPLICIT SEQUENCE{
                        rpn1    RPNStructure,
                        rpn2    RPNStructure,
```

```
        op                      [46] CHOICE{
                        and             [0]  IMPLICIT NULL,
                        or              [1]  IMPLICIT NULL,
                        and-not  [2]  IMPLICIT NULL} }}
AttributesPlusTerm ::= [102] IMPLICIT SEQUENCE{
                attributes      AttributeList,
        term    Term}
        END
```