ANSI/NISO Z39.50-1995

Information Retrieval (Z39.50): Application Service Definition and Protocol Specification

__Status__: This is the Z39.50 Maintenance Agency Official Text for Z39.50-1995. (The Maintenance Agency for Z39.50 is the Library of Congress.) This text is submitted to NISO for publication; the ANSI/NISO publication may differ in style and there may be minor editorial differences, but otherwise the content should be identical. If there are changes to this standard due to latent defects or approved enhancements, they will be carefully documented. See http://lcweb.loc.gov/z3950/agency.

Abstract

This standard specifies a client/server based protocol for Information Retrieval. It specifies procedures and structures for a client to search a database provided by a server, retrieve database records identified by a search, scan a term list, and sort a result set. Access control, resource control, extended services, and a "help" facility are also supported. The protocol addresses communication between corresponding information retrieval applications, the client and server (which may reside on different computers); it does not address interaction between the client and the end-user.

Z39.50 Maintenance Agency
Official Text for Z39.50-1995

July 1995

Contents

Foreword ii
1. Introduction 1
2. Definitions 2
3. Information Retrieval Service 6
3.1 Model and Characteristics of the Information Retrieval Service 6
3.2 Facilities of the Information Retrieval Service 9
3.2.1 Initialization Facility 9
3.2.2 Search Facility 12
3.2.3 Retrieval Facility 16
3.2.4 Result-set-delete Facility 18
3.2.5 Access Control Facility 19
3.2.6 Accounting/Resource Control Facility 20
3.2.7 Sort Facility 24
3.2.8 Browse Facility 25
3.2.9 Extended Services Facility 26
3.2.10 Explain Facility 30
3.2.11 Termination Facility 38
3.3 Message/Record Size and Segmentation 39
3.4 Operations and Reference-id 43
3.5 Concurrent Operations 44
3.6 Composition Specification 44
3.7 Type-1 and Type-101 Queries 46
4. Protocol Specification 49
4.1 Abstract Syntax and ASN.1 Specification of Z39.50 APDUs 49
4.2 Protocol Procedures 63
4.3 Rules for Extensibility 72
4.4 Conformance 72

Appendices

1. OID: Z39.50 Object Identifiers 78
2. CTX: Application Context basic-Z39.50-ac 80
3. ATR: Attribute Sets 81
4. ERR: Diagnostic Errors 85
5. REC: Record Syntaxes 92
6. RSC: Resource Report Formats 112
7. ACC: Access Control Formats 114
8. EXT: Extended Services Defined by this Standard 117
9. USR: User Information Formats 129
10. ESP: Element Specification Formats 131
11. VAR: Variant Sets 133
12. TAG: Schema and TagSet Definitions 136
13. ERS: Extended Result Set Model 139
14. RET: Z39.50 Retrieval 141
15. PRO: Z39.50 Profiles 156

Foreword

(This foreword is not a formal part of American National Standard ANSI/NISO Z39.50-1995, but is included for information only.)

ANSI Z39.50-1995, *Information Retrieval (Z39.50) Application Service Definition and Protocol Specification* is a revision of ANSI Z39.50-1992.

Editors note: Drafts versions of this standard were referred to as Z39.50-1994. This was changed to Z39.50-1995, as part of the approval and publication process. There is no approved 1994 version of Z39.50. Z39.50-1995 is the final, approved version of the standard for which the various drafts were referred to as Z39.50-1994. Implementors should take note that any earlier draft, referred to as Z39.50-1994, is not the latest version of this standard.

The 1992 version was a revision of Z39.50-1988, which was prepared by a NISO (National Information Standards Organization) committee that was disbanded after Z39.50-1988 was approved. In its place the Z39.50 Maintenance Agency was established in 1989, administered at the Library of Congress.

The protocol was originally proposed (in 1984) for use with bibliographic information. As interest in Z39.50 broadened, the Z39.50 Implementors Group (ZIG) was established, in 1990. Members include manufacturers, vendors, consultants, information providers, and universities, who wish to access or provide access to various types of information, including bibliographic, text, image, financial, public utility, chemical, and news. ZIG membership is open to all interested parties.

Various enhancements were proposed by implementors for the 1992 version, to support a wide range of information retrieval activities. But those features were not yet fully developed, and their incorporation into the 1992 standard would have caused significant delay. The Z39.50 Maintenance Agency had been assigned, as top priority, to revise Z39.50-1988 to achieve bit-compatibility with the international standard, ISO 10162/10163, *Search and Retrieve*, SR. (Z39.50-1992 replaced and superseded Z39.50-1988, and is a compatible superset of SR.) The proposed new features were therefore deferred, with a commitment to implementors that development of the required features would proceed, and that the resultant subsequent version would be a compatible superset of the 1992 standard.

In 1992 the maintenance agency conducted a formal survey among Z39.50 implementors to determine the relative importance of proposed new features. The survey's purposes were to begin to narrow the list to a manageable set, to determine whether the proposed features were adequately specified and understood, and to gauge their perceived cost and complexity. The survey results revealed certain features to be indispensable, and that certain others features could be eliminated from further consideration. For a third set of features, the survey was inconclusive and the disposition of those features eventually was determined by consensus.

Development of Z39.50-1995 began in late 1991. For each meeting of the ZIG, from December 1991, through April 1994, a revised draft was developed by the Z39.50 Maintenance Agency. Each draft underwent careful scrutiny by implementors, and was discussed at length both over the ZIG Internet mail list, and at the ZIG meeting. Comments and discussion for each draft, and agreements reached at each ZIG meeting, were incorporated into the subsequent draft. In April 1994, the ZIG recommended that the draft be finalized.

The 1992 version came to be known as "version 2", and the 1995 version, "version 3". However, although these version designations do have specific *protocol* significance, they do not refer to versions of the *standard*. Z39.50-1992 specifies protocol version 2; Z39.50-1995 specifies protocol versions 2 and 3.

Although Z39.50-1992 replaced and superseded Z39.50-1988 (and Z39.50-1988 is obsolete) the relationship between Z39.50-1992 and Z39.50-1995 is quite different: Z39.50-1995 is a compatible superset of the 1992 version. An implementor may obtain complete details of version 2 from the Z39.50-1995 document, and build an implementation compatible with Z39.50-1992.

Z39.50-1995 represents a consensus of the ZIG, which has in effect acted in an advisory role to the maintenance agency, in the effort to develop both Z39.50-1992 and the Z39.50-1995.

Basics of the Protocol

The protocol specifies formats and procedures governing the exchange of messages between a client and server enabling the client to request that the server search a database and identify records which meet specified criteria, and to retrieve some or all of the identified records.

The client may initiate requests on behalf of a user; the protocol addresses communication between corresponding information retrieval applications, the client and server (which may reside on different computers); it does not address interaction between the client and user.

Z39.50-1992 provides the following basic capabilities, all of which are supported in Z39.50-1995 as well. The client may send a search, indicating one or more databases, and including a query as well as parameters which determine whether records identified by the search should be returned as part of the response. The server responds with a count of records identified and possibly some or all of the records. The client may then retrieve selected records. The client assumes that records selected by the search form a "result set" (an ordered set, order determined by the server), and records may be referenced by position within the set. Optional capabilities include:

- The client may specify an *element set* indicating data elements to retrieve in cases where the client does not wish to receive complete database records. For example, the client might specify "If 5 or less records are identified, transmit ’full’ records; if more than 5 records are found, transmit ’brief’ records".
- The client may indicate a *preferred syntax* for response records, for example, USMARC.
- The client may *name* a result set for subsequent reference.
- The client may *delete* a named result set.
- The server may impose *access control* restrictions on the client, by demanding authentication before processing a request.
- The server may provide *resource control* by sending an unsolicited or solicited status report; the server may suspend processing and allow the client to indicate whether to continue.

Query Formulation

This standard fully specifies and mandates support of the *type-1* query, expressed by individual search terms, each with a set of attributes, specifying, for example, type of term (subject, name, etc.), whether it is truncated, and its structure. The server is responsible for mapping attributes to the logical design of the database. Terms may be combined in a type-1 query, linked by boolean operators. Terms and operators are expressed in Reverse Polish Notation.

Attribute Sets

The attributes associated with a search term belong to a particular attribute set, whose definition is *registered*, that is, assigned a unique and globally recognized *attribute-set-id*, an *Object Identifier*, which is included within the query.

Appendix ATR defines and registers the attribute-set bib-1, which specifies various attributes useful for bibliographic queries. Additional attribute sets may be registered outside of the standard. The bib-1 attribute set was developed by the bibliographic community; it is intended that attribute sets will be developed and registered as needed by other communities.

Response Records

The protocol distinguishes two types of records that may occur in response messages from the server: database and diagnostic records.

Appendix REC registers object identifiers for various MARC formats, including USMARC, UKMARC, Norway MARC and CANMARC; these object identifiers accompany database records returned by the server. There are several other types of record formats defined, and there is a provision for registration of additional record formats.

Diagnostic records are similarly accompanied by an object identifier which identifies their format. Appendix ERR defines and registers two diagnostic record formats (one of which was defined in Z39.50-1992) which includes various diagnostic codes useful for bibliographic applications. Additional diagnostic record formats may be registered.

New Features

Provided below is a summary of the enhancements in Z39.50-1995. The designations "version 2" and "version 3" refer to protocol version; "Z39.50-1992" and "Z39.50-1995" refer to the respective standards. Thus where a particular feature is described as "new in Z39.50-1995", that generally means it applies in either protocol version. An example is Scan: an implementor may add the Scan service to an existing implementation of Z39.50-1992 without incorporating any other new features.

The enhancements described below fall into four categories: search, retrieval, new services and facilities, and miscellaneous enhancements.

Search

Attributes

There are a number of enhancements pertaining to attributes and attribute sets. In version 3, attributes may be combined from different attribute sets, within a single query (even for a single search term). This presents two advantages: First, it is useful when searching multiple databases. (Although version 2 supports multiple-database searches, all attributes within a query must belong to a single attribute set, which inhibits the ability to search multiple databases, unless those databases are similar.) Second, new attribute sets may now be defined with less replication.

Version 3 provides two further enhancements allowing flexibility in the definition of attribute sets. First, new data types for attribute values are defined (in version 2 only numeric values are allowed). Second, an attribute set definition may now list alternative sets of evaluation rules (for example, whether the server is allowed to substitute an attribute that it thinks is more appropriate), and the query may select one of the alternatives. The enhanced bib-1 attribute set definition exploits this new feature.

The bib-1 definition in Z39.50-1995 also includes many new attributes (as well as all of the attributes in Z39.50-1992).

Extended Result Set Model

The basic model of a result set is developed in Z39.50-1992; the 1995 version describes an "extended result set model", which supports extended proximity searching.

The extended model also supports a new version 3 search function, *restriction*, which is (in effect) an operation on a result set. It permits selection of records from a result set, based on specified attributes.

Search Term

The search term for a query may take on a variety of data types in version 3. (In version 2 a search terms is binary and thus essentially has no data type, so the type is often described by a structure attribute.) This enhancement will simplify queries (as well as attribute set definitions) by reducing the need for structure attributes.

Intermediate Results

In Z39.50-1995 the server may provide information per query *component* (i.e. per sub-query, per database), as part of the Search response (version 3 only), or as part of resource-control when the server reports on the progress of the search. The server may also create and provide access to a result set for individual query components.

Retrieval

Segmentation

In version 2, a retrieval response is limited to a single message; the server attempts to fit the requested records into the message, and if it cannot, it simply fits as may as it can. The client might want to retrieve, for example, ten thousand records, knowing it cannot retrieve them in a single message. Typically the client will request all ten thousand records, wait for the response, determine how many records are retrieved, and then send another request for the remaining records. This works well in many environments but is unacceptably slow for high-speed networks. The server must await a request before sending each set of records, which introduces a delay; the delay may be negligible for conventional networks, but is intolerable for high-speed networks. In version 3 a server may respond to a retrieval request with multiple consecutive response messages without intervening requests.

A more serious segmentation problem occurs when a *single* record is too large to fit in a single message. Version 3 thus introduces a second level of segmentation: an individual record may span response messages. A client or server may choose to support either level of segmentation, or no segmentation (in which case version 2 rules apply).

Retrieval Tools

The ZIG has worked intensively over two years to develop an extensive model and suite of tools for a wide range of retrieval functions to support various retrieval applications, in particular, document retrieval. The model is detailed in Appendix RET. Several new object classes are designated in Z39.50-1995 (schemas, tagSets, variants) and specific objects from these and other classes are defined. Appendix RET provides detailed semantics for these objects and describes how they are used together to provide a variety of document retrieval capabilities. Following are a few examples:

- A single database record might include a number of documents. The client may discover and retrieve a specific document, rather than the entire database record.
- The client may retrieve a specific portion of a document, logical or physical, for example, specific pages, a specific chapter, a specific

caption, all captions, or all images. The client might retrieve just *headings*, for example, all chapter or section headings.

- A document might be available in a wide variety of formats (e.g. postScript, SGML), languages, presentation parameter (e.g. line length, lines per page, columns), and other variants. The client may discover what variants are supported for a document, as well as information associated with a particular variant form: for example the cost to retrieve the document according to a specific variant, or its size. Finally, the client may then retrieve the document (or specific portion) according to the desired variant.
- Associated with a document, for a given search, may be *hits*: pointers to terms (within the document) relevant to the search. The client might retrieve hits along with a document to quickly locate the satisfying portions. Or the client might retrieve only the hits (ranked in order of importance), and subsequently retrieve only the indicated satisfying portions.

New Services and Facilities

Scan and Sort

Scan and Sort are new services in Z39.50-1995. These are used respectively to scan terms in a list or index, and to sort a result set.

Scan is currently the only service in the Z39.50 Browse facility, but it is intended that various other browse capabilities will be added in future versions.

Extended Services

Extended Services is a new facility in Z39.50-1995. It includes a new Z39.50 service, the *Extended Services service*, used to initiate a specific extended service task, which is executed outside of the Z39.50 session and whose progress may be monitored using Z39.50 services. Specific extended services include: save a result set, set a periodic query schedule, export a document, order a document, and update a database.

Explain

The new Explain facility allows a client to retrieve details of the server implementation: general features (description, contact information, hours of operation, restrictions, usage cost, etc.) databases available for searching, indexes, attribute sets, attribute details, schemas, record syntaxes, sort capabilities and extended services. The server maintains Explain information in a special database that may be accessed by the client using the Z39.50 search and retrieval facilities. The format of the Explain information is detailed in the standard.

Some Explain information is transparent to the client, intended for direct display to the client-user, and is so designated (e.g. "general features"). Some Explain information is intended to be shared by client and user. For example, the client may retrieve a list of searchable databases; for each database in the list the client might display an *informal* name, an icon, and a brief description. Meanwhile the client would retain the *actual* database name to be used in a protocol message, which probably would not be displayed. Some Explain information may be completely transparent to the user. For example, the client may retrieve information about attributes supported for a database and use that information when formulating a query (when converting a user-supplied query to a Z39.50 type-1 query).

Miscellaneous Enhancements

Termination and Re-initialization

Version 3 includes a more flexible approach to termination of a Z39.50 session, to allow, in effect, re-initialization without taking down the network connection.

Concurrent Operations

Multiple concurrent operations are allowed in version 3. In version 2, operations are strictly serial.

Diagnostics

Most Z39.50 services include diagnostic capability. In version 2 a diagnostic must conform to a specific format defined within the standard. In version 3, diagnostic formats may be externally defined and registered. One such (new) format is defined, along with a comprehensive set of diagnostics.

Access Control Formats

Z39.50-1992 provides access control, but does not define any access control formats. Z39.50-1995 defines formats for encryption and authentication, and a format allowing the server to prompt the client for arbitrary information.

Character Set Support

A new data type, "International String", has been introduced for character strings. Its definition allows greater flexibility for a client and server to agree to the use of a particular language and one or more character sets during a session.

Units

New data types are introduced for support of units. These definitions allow standard representations to be used to represent unit type and unit. For example, unit type might be "mass", and unit, "kilogram".

Extensibility and Negotiation

Version 3 provides a powerful extensibility feature. Each protocol message includes a field designated for information whose format is to be defined externally. These externally defined formats will be registered and maintained by the Z39.50 Maintenance Agency, as provisional extensions to the standard, for experimental use and possible consolidation into a subsequent version.

In Z39.50-1995 the concept of a "negotiation record" is introduced. The client may include a negotiation record within the initialization message to propose that some condition be in effect for the session (for example, the use of a particular language and one or more character sets). The server may respond, indicating whether the proposal is accepted, or indicate a counter-proposal.

The negotiation record is an application of the new extensibility feature. Negotiation records will be defined externally and maintained by the Z39.50 Maintenance Agency.

Information Retrieval (Z39.50): Application Service Definition and Protocol Specification

1. Introduction

This standard, ANSI/NISO Z39.50-1995, *Information Retrieval (Z39.50) Application Service Definition and Protocol Specification*, is one of a set of standards produced to facilitate the interconnection of computer systems. It is positioned with respect to other related standards by the Open Systems Interconnection (OSI) basic reference model (ISO 7498). This standard defines a protocol within the application layer of the reference model, and is concerned in particular with the search and retrieval of information in databases.

1.1 Scope and Field of Application

This standard describes the Information Retrieval Application Service (section 3) and specifies the Information Retrieval Application Protocol (section 4). The service definition describes services that support capabilities within an application; the services are in turn supported by the Z39.50 protocol. The description neither specifies nor constrains the implementation within a computer system. The protocol specification includes the definition of the protocol control information, the rules for exchanging this information, and the conformance requirements to be met by implementation of this protocol.

This standard is intended for systems supporting information retrieval services, for organizations such as information services, universities, libraries, and union catalogue centers. It addresses connection-oriented, program-to-program communication. It does not address the interchange of information with terminals or via other physical media.

1.2 Version

This standard, Z39.50-1995, specifies versions 2 and 3 for the Z39.50 service and protocol.

Notes:

1. ANSI Z39.50-1992 specifies version 2 only.
2. For compatibility with version 1 of the Search and Retrieve Protocol (ISO 10163-1991), version 2 of Z39.50 is assumed identical to version 1 of Z39.50; thus implementations that support version 2 automatically support version 1. (Version 1 of ANSI Z39.50-1992 should not be confused with ANSI Z39.50-1988.)

Certain procedures specified within the standard apply specifically to version 2 or version 3 and are noted as such.

1.3 References

ANSI/NISO Z39.53-1994 -- *Codes for the Representation of Languages for Information Interchange.*

ANSI/NISO Z39.58-1992 -- *Common Command Language for Online Interactive Information Retrieval.*

ISO 2709 -- *Documentation - Format for Bibliographic Information Interchange on Magnetic Tape* 1981.

ISO 4217 -- *Codes for the representation of currencies and funds* 1990.

ISO 7498 -- *Information Processing Systems - Open Systems Interconnection - Basic Reference Model* 1984.

ISO 8649 -- *Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element* 1987.

ISO 8650 -- *Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element* 1987.

ISO 8777 -- *Information and Documentation - Commands for Interactive Text Searching.*

ISO 8822 -- *Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Service Definition* 1988.

ISO 8824 -- *Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)* 1990.

ISO 8825 -- *Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)* 1990.

ISO 10160 -- *Information and Documentation - Interlibrary Loan Application Service Definition for Open Systems Interconnection* 1991.

ISO 10161 -- *Information and Documentation - Interlibrary Loan Application Protocol Specification for Open Systems Interconnection* 1991.

ISO 10163 -- *Information and Documentation - Search and Retrieve Application Protocol Specification for Open Systems Interconnection* 1991.

ISO -- *International Register of Coded Character Sets To Be Used with Escape Sequences* 1992.

2. Definitions

A-association -- See **Application association**.

Abstract database record--An abstract representation of the information in a database record. An abstract database record may be formed by the application of an abstract record structure (defined by a schema) to the database record. An element specification may be applied to an abstract database record forming another instance of the abstract database record.

Abstract record structure -- The primary component of a database schema. An abstract record structure applied to a database record results in an abstract database record.

Abstract syntax -- A description of a particular data type using an abstract syntax notation. It can be referenced by an OID (object identifier).

Abstract syntax notation -- A language that allows the denotation of data types in a representation-independent manner. ASN.1 is an example.

Access point -- A unique or non-unique key that can be specified either singly or in combination with other access points in a search for records. An access point may be equivalent to an element (defined by an abstract syntax), derived from a set of one or more elements, or unrelated to any element.

Access point clause -- An operand of a type-1 query (informal).

Aggregate present response -- Segment requests (if any) together with the Present response, for a Present operation.

APDU -- See **Application Protocol Data Unit**.

Application association -- A communication session between a database user and a database provider. It may consist of one or more consecutive Z-associations.

Application Protocol -- The rules governing the format and exchange of information between an origin and target.

Application Protocol Data Unit -- A unit of information, transferred between origin and target, whose format is specified by the Z39.50 protocol, consisting of application-protocol-information and possibly application-user-data.

Application Protocol Control Information -- Information conveyed by an application protocol data unit.

AppliedVariant -- One of three usages for a variant specification. The applied variant is the variant specification that the target applied to an element included in a retrieval record. See also **variantRequest** and **supportedVariant**.

ARS -- See **Abstract record structure.**

ASN.1 -- Abstract Syntax Notation One, as specified in ISO 8824 and ISO 8825.

Attribute -- A characteristic of a search term, or one of several characteristic components which together form a characteristic of a search term.

Attribute element -- An attribute represented by a pair of components: an attribute type and a value of that type.

Attribute list -- A set of attribute elements and the attribute set id to which it belongs. An attribute list is combined with a search term to form an operand of a type-1 query. Usually, one of the attribute elements from the set corresponds to a normalized access point, against which the term (as qualified by the other attribute elements) is matched.

Attribute set -- A set of attribute types, and for each, a list of attribute values. Each type is represented by an integer, unique within that set (as identified by its attribute set id), and each value for a given type is unique within that type.

Attribute set id -- An OID that identifies an attribute set, to which an attribute element (within an attribute list) belongs.

Attribute type -- A component of an attribute element. An attribute set defines one or more attribute types and assigns an integer to each type (it also defines values specific to each type). For example, bib-1 assigns the integer 1 for the attribute type "Use."

Attribute value -- A component of an attribute element. An attribute set defines one or more values for each attribute type that it defines. For example, bib-1 defines the Use attribute "personal name."

Client -- The application that includes the origin; the database user.

Client system -- The system on which the client resides.

Composition specification -- A specification that may be included in a Present request to indicate the desired composition (elements and record syntax) of the retrieval records. It includes a schema identifier, element specification, and record syntax identifier.

Conditionally confirmed service -- A service that may be invoked as confirmed or non-confirmed. It is defined in terms of a request (from the origin or target) followed possibly by a response (from the peer). For example, Resource-control is a conditionally confirmed service, initiated by the target. See also **Non-confirmed service** and **Confirmed service**.

Confirmed service -- A service that is defined in terms of a request (from the origin or target) followed by a response (from the peer). For example, Search is a confirmed service, initiated by the origin; Access-control is a confirmed service initiated by the target. See also **Non-confirmed service** and **Conditionally-confirmed service**.

Database -- A collection of information units containing related information. Each unit is a database record.

Database record -- A local data structure representing an information unit in a database.

Database schema -- A common understanding shared by the origin and target of the information contained in the records of the database, which allows the subsequent selection of portions of that information via an element specification. A schema defines an abstract record structure, which, when applied to a database record, results in an abstract database record.

Data element -- See **Element.**

Element -- A unit of information defined by a schema.

ElementRequest -- A request, included with an element specification, for the retrieval of a specific element. The element request may include a variantRequest, indicating the desired variant form of the element.

Element set name -- An element specification in the form of a primitive name.

Element specification -- An instance of an element specification format, or an element set name. An element specification transforms an abstract database record into another instance of the abstract database record (this may be a null transformation). The element specification selects elements from the abstract database record, and possibly also specifies variant forms for those elements.

Element specification format -- A structure used to express an element specification.

Element specification identifier -- The object identifier of an element specification format, or an element set name.

Exceptional record size -- The maximum size of the record that may be included in a Present response, in the special case when a single, exceptionally large record (i.e. larger than preferred-message-size) is requested.

Facility -- A logical group of Z39.50 services; in some cases, a single service. For example, the Retrieval facility consists of the Present service and the Segment service; the Search facility consists of the Search service. Alternatively, a facility might not consist of services, but instead might use services of other facilities. For example, the Explain facility does not define any services, but uses the Search and Present services.

Final fragment -- A fragment that ends at the end of a record. See **Fragment.**

Fragment -- A proper substring of a record. (This definition is meaningful only in the context of level-2 segmentation, described in section 3.3.3; within that section, a record is considered to be a string of bytes.)

GRS -- Generic Record Syntax.

Initiating request -- A request that initiates an operation.

Intermediate fragment -- A fragment that neither starts at the beginning nor ends at the end of a record. See **Fragment**.

IR -- Information Retrieval.

Item -- (1) A result set item. (2) A bibliographic item; see ISO 10160.

Maximum segment size -- The largest allowable segment of an aggregate Present response (when segmentation is in effect).

Name -- A linguistic construct, expressed in some language, that corresponds to an object. A name denotes (i.e. identifies) the object to which it is bound.

Non-confirmed service -- A service that is defined in terms of a request from the origin or target, with no corresponding response. For example, Segment is a non-confirmed service initiated by the target. See also **Confirmed service**.

Object identifier -- An unambiguous, globally-recognized, registered identifier for a data object, assigned by a registration authority.

OID -- See **Object identifier**.

Operation -- An initiating request and the corresponding terminating response, along with intervening related messages. For example, a Search operation always includes a Search request and Search response, and may also include access control and resource control messages. Multiple concurrent operations may occur within a Z-association.

Operation type -- The name of an initiating request. For example a Search request initiates an operation whose type is "search."

Origin -- The entity that initiates a Z-association and initiates operations during the Z-association.

Origin service-user -- That portion of a client that makes requests upon the origin. See **Service-user**.

OSI -- Open Systems Interconnection.

P-context -- See **Presentation context**.

Preferred message size -- The maximum size of a Search response or Present response when no segmentation is in effect. It is expressed in terms of the sum of the sizes (in bytes) of the response records, not including protocol control information.

Presentation context -- The pairing of an abstract syntax with a transfer syntax, negotiated by the presentation layer, in order for that abstract syntax to be used during the application association.

Primitive -- See **Service primitive**.

Primitive Name -- A name whose internal structure is not required to be understood or have significance to users of the name.
Note: **primitive name** is not related to **primitive**.

Record syntax--An abstract syntax requested by the origin or used by the target to represent retrieval records. For a complete definition, see section 3.6.3.

Response record -- A retrieval record or a surrogate diagnostic record, representing a database record, in a Search response or (aggregate) Present response.

Result set -- A local data structure used as a selection mechanism for the transfer of records, identified by a query. Its logical structure is a named, ordered list of result set items, and possibly, unspecified information which may be used as a surrogate for the search that created the result set.

Result set item -- A database name, a pointer to a record within the database, and possibly, additional, unspecified information associated with the record.

Result set record -- An idiomatic expression referring to the database record represented by a result set item. See **Result set**.

Retrieval record -- The exportable structure defined by the application of a record syntax to an abstract database record.

RPN query -- A search query represented in reverse polish notation (RPN) format.

Schema -- See **Database schema**.

Segment -- A message that is sent (or is in preparation for transmission) by the target as part of an aggregate Present response, i.e. a Segment request or Present response.

Server -- The application that includes the target; the database provider.

Server system -- The system on which the server resides.

Service -- (1) A Z39.50 service, as in the "search" service; (2) an extended service, as in the "persistent result set extended service"; (3) the service-provider.

Service primitive -- An abstract, implementation-independent representation of an interaction between the service-user and the service-provider. The four types of service primitives are: Request, Indication, Response, and Confirmation.

Service-provider -- An abstraction of the totality of those entities (the origin and target) that provide a service to peer service-users. The concept of service-provider is employed to facilitate the specification of protocol procedures. It is used only within 4.2.2, to describe the protocol model.
Note: the service-provider is not related to the database provider nor to the provider of telecommunication services.

Service-user -- An origin service-user or a target service-user. That portion of a client or server that makes requests upon the origin or target respectively. The concept of service-user is employed to facilitate the specification of protocol procedures. It is used only within 4.2.2, to describe the protocol model.
Note: The service-user is not related to the database user.

Simple Present response -- An aggregate Present response consisting of a single segment, i.e., consisting of a Present response only, and no Segment requests.

Starting fragment -- A fragment that starts at the beginning of a record. See **Fragment**.

SupportedVariant -- One of three usages for a variant specification. A supportedVariant is a variant specification that the target lists as supported for a particular element. See also **appliedVariant** and **variantRequest**.

Surrogate diagnostic Record -- A diagnostic record supplied in place of a retrieval record, representing a database record.

Tag -- The identifier of an element (or of a node of the tagPath representing an element). It consists of a tagType and a tagValue.

TagPath -- A sequence of nodes from the root of a tree to the node that the tagPath represents (when the elements of a record are represented hierarchically, as a tree). Each node of a tagPath is represented by a tag. The end-node might be a leaf-node, in which case the tagPath represents an element; otherwise the tagPath represents the subtree whose root is that node.

TagSet -- The tagValues (and recommended data types) for a set of elements.

TagSetId -- an object identifier serving as a persistent identifier for a tagSet.

TagType -- A short-hand (integer) identifier for a tagSet. A schema definition may assign a tagType to a TagSetId, to identify a particular tagSet (within the context of the schema definition).

TagValue -- The identifier of an element (or of a node of the tagPath representing an element). It may be either integer or string, and it is qualified by a tagType.

Target -- The entity that accepts a Z-association.

Target service-user -- That portion of a server that makes requests upon the target. See **Service-user**.

Terminating response -- A response that ends an operation.

Transfer syntax -- A syntax that when paired with an abstract syntax forms a record syntax.

Triple -- A 3-tuple. (I.e. an n-tuple, where n = 3.)

Type-1 query -- See **RPN Query**.

Variant -- One of possibly several forms in which an element is available for retrieval. The origin may request, or the target present, an element according to a specific variant. The target may indicate what variants are available for an element.

Variant list -- A list provided by the target of the supportedVariants for a particular element.

VariantRequest -- One of three usages for a variant specification. A variantRequest is a variant specification occurring within an element request. See also **appliedVariant** and **supportedVariant**.

Variant set -- A definition of a set of classes; for each class, a set of types; and for each type, a set of values. A variant specification consists of a set of variantSpecifiers from a particular variant set.

Variant set identifier -- An OID identifying a variant set.

Variant Specification -- A variantRequest, applied-Variant, or supportedVariant. A variant specification is a sequence of triples, each of which is a variantSpecifier.

Variant Specifier -- A component of a variant specification. It consists of a class, a type defined for that class, and a value defined for that type.

Z-association -- See **Z39.50-association**.

Z39.50-association -- A session, explicitly established by the origin and either explicitly terminated by the origin or target, or implicitly terminated by termination of the A-association. Communication between origin and target is via a Z39.50-association within an application association. There may be multiple, consecutive Z-associations within an A-association.

3. Information Retrieval Service

The Information Retrieval service definition describes an activity between two applications: an initiating application, the client, and a responding application, the server. The server is associated with one or more databases.

Communication between the client and server is carried out by the Z39.50 protocol, which is specified in section 4. The specification is logically divided into procedures pertaining to the client and procedures pertaining to the server. The portions of the client and server that carry out the Z39.50 protocol procedures are referred to respectively as the Z39.50 origin and the Z39.50 target.

3.1 Model and Characteristics of the Information Retrieval Service

Communication between origin and target is via a Z39.50-Association (Z-association) within an application association (A-association; see 4.2.1.2). A Z-association is explicitly established by the origin and may be explicitly terminated by either origin or target, or implicitly terminated by termination of the A-association.

There may be multiple consecutive Z-associations within an A-association. There may be multiple consecutive as well as concurrent operations (see 3.1.2) within a Z-association.

The roles of origin and target may not be reversed within a Z-association. A Z-association cannot be restarted, thus once a Z-association is terminated no status information is retained, except information that is explicitly saved.

The service definition describes services and operations; models for these are described in 3.1.1 and 3.1.2. Services are grouped by facilities; the Z39.50 facilities and services are defined in 3.2.

3.1.1 Z39.50 Services

Z39.50 services are carried out by the exchange of messages between the origin and target. A message is a request or a response. Services are defined to be confirmed, non-confirmed, or conditionally-confirmed.

A confirmed service is defined in terms of a request (from the origin or target) followed by a response (from the peer). For example, Search is a confirmed service, initiated by the origin; the Search service is defined in terms of a Search request from the origin followed by a Search response from the target. Access-control is an example of a confirmed service initiated by the target.

A non-confirmed service is defined in terms of a request from the origin or target, with no corresponding response. For example, TriggerResourceControl is a non-confirmed service initiated by the origin; Segment is a non-confirmed service initiated by the target.

A conditionally-confirmed service is a service that may be invoked as either a confirmed or non-confirmed service. It is defined in terms of a request (from the origin or target) followed possibly by a response (from the peer). For example, Resource-control is a conditionally-confirmed service, initiated by the target.

3.1.2 Z39.50 Operations

This standard describes eight operation types: Init, Search, Present, Delete, Scan, Sort, Resource-report, and Extended-services.

A request from the origin of a particular operation type initiates an operation of that type (for example a Search request initiates a Search operation) which is terminated by the respective response from the target. Only the origin may initiate an operation, and not all origin requests do so (see 3.4).

A request that initiates an operation is called an initiating request and a response that ends an operation is called a terminating response. From the origin perspective, an operation begins when it issues the initiating request, and ends when it receives the terminating response. From the target perspective, the operation begins when it receives the initiating request and ends when it sends the terminating response. An operation consists of the initiating request and the terminating response, along with any intervening related messages (see 3.4).

3.1.3 Model of a Database

The objective of this standard is to facilitate the open interconnection of clients and servers for applications where clients search and retrieve information from server databases. The ways in which databases are implemented differ considerably; different systems have different styles for describing the storage of data and the means by which it can be accessed. A common, abstract model is therefore used in describing databases, to which an individual system can map its implementation. This enables different systems to communicate in standard and mutually understandable terms, for the purpose of searching and retrieving information from a database. The search and retrieval models are described in 3.1.4 and 3.1.5.

The term database, as used in this standard, refers to a collection of records. Each record is a collection of related information, treated as a unit. The term database record refers to a local data structure representing the information in a particular record. Associated with a database are one or more sets of access points that can be specified in a search for database records (see 3.1.4), and one or more sets of elements that may be retrieved from a database record (see 3.1.5). An access point is a unique or non-unique key that can be specified either singly or in combination with other access points in a search for records. An access point may, but need not, be related to an element; it can be equivalent to an element, derived from a set of one or more elements, or unrelated to any element.

3.1.4 Searching a Database

A query is applied to a database, specifying values to be matched against the access points of the database. The subset of records formed by applying a query is called the result set (see 3.1.6). A result set may itself be referenced in a subsequent query and manipulated to form a new result set.

A search request specifies one or more databases and includes a query. The type-1 query defined in this standard (see 3.7) consists of either a single access point clause, or several access point clauses linked by logical operators. For example:

> In the database named "Books" find all records for which the access point ’title word’ contains the value "evangeline" AND the access point ’author’ contains the value "longfellow."

Each access point clause consists of a search term and attributes. The attributes qualify the term; usually, one of the attributes corresponds to a normalized access point, against which the term (as qualified by the other attributes) is matched. Each attribute is a pair representing an attribute type and a value of that type (for example, type might be "access point" and value "author"; or type might be "truncation" and value "left").

Each attribute is qualified by an attribute set id which identifies the attribute set to which the attribute belongs. An attribute set specifies a set of attribute types, and for each, a list of attribute values.

3.1.5 Retrieving Records from a Database

Following the processing of a search, the result set is made available by the target, to the origin, for subsequent retrieval requests. When requesting the retrieval of a record from a result set, the origin may supply a database schema identifier, element specification, and record syntax identifier.

For the purpose of retrieving records from a result set, associated with each database are one or more schemas. A schema represents a common understanding shared by the origin and target of the information contained in the records of the database, to allow the subsequent selection of portions of that information via an element specification.

A schema defines an abstract record structure which, when applied to a database record results in an abstract database record, which is an abstract representation of the information in the record. An element specification applied to an abstract database record results in another instance of the abstract database record (the latter may be a null transformation). The element specification selects elements from the abstract database record, and may also specify variant forms for those elements.

The target applies a record syntax to an abstract database record, resulting in an exportable structure referred to as a retrieval record.

3.1.6 Model of a Result Set

In general, it is assumed that query processing does not necessarily require physical access to records; a result set is thus assumed to be the identification of (e.g., pointers to) records, as opposed to the actual set of records, selected by a query. (It is not assumed that the database records are locked. Methods of concurrency control, which would prevent modification or deletion of result set records, are not addressed by this standard.) A result set may be used as a selection mechanism for the transfer of records between systems; the result set itself is considered to be a purely local data structure and is not transferred (that is, records are transferred, but not the local pointers to the records).

For the purpose of retrieving records, the logical structure of a result set is that of a named, ordered list of items. Each item is a triples consisting of:

(a) an ordinal number corresponding to the position of the triple in the list,
(b) a database name, and
(c) a unique identifier (of local significance only) of a record within the database named in (b).

A result set item is referenced by its position within the result set, that is, by (a).

For the purpose of searching, when a result set is used as an operand in a query, the logical structure is one of the following:

• Basic model	A set of pairs, each consisting of (b) and (c) of the above model for retrieval.
• Extended model	A set of triples, each consisting of (b) and (c) of the retrieval model; and including unspecified information associated with each record, which may be used as a surrogate for the search that created the result set.

Note: Query specifications may indicate that the basic model applies, or under what condition the extended model applies and the nature of the unspecified information. For the type-1 query, when version 2 is in effect, the basic model applies.

3.1.7 Model of Extended Services

The family of Z39.50 services includes the Extended Service (ES) service. "Extended services" refers to a class of services recognized by this standard, but which are not Z39.50 services (as described in 3.1.1). The ES service *is* a Z39.50 service, and an ES operation results in the initiation of an extended services task. The *task* is *not* considered part of the Z39.50 ES operation.

An ES operation is initiated by the origin, via an ES request. The ES response, which completes the operation, does not (necessarily) signal completion of the task; it may indicate for example that the task has started or is queued (or it might indicate that the task has been completed; in fact the ES request may specify that the task should be completed prior to the ES response). An ES task may have a lifetime beyond the Z-association.

Examples of extended services are: saving a result set or query, and exporting or ordering a document.

Each ES task is represented by a database record, called a task package, maintained by the target in a special database, the "extended services database". The origin uses the ES request to cause creation of a task package on the ES database. The database may be searched, and records retrieved, by the Z39.50 Search and Retrieval facilities. The origin may search for packages of a particular type, or created by a particular user, or of a particular status (i.e., pending, active, or complete), or according to various other criteria. In particular, the origin may search the database after submitting an ES request (during the same or a subsequent Z-association), for a resulting task package, to determine status information pertaining to the task, for example, to determine whether the task has started.

3.1.8 Explain

The origin may obtain details of the target implementation, including databases, attribute sets, diagnostic sets, record syntaxes, and element specifications supported. The origin obtains these details through the Z39.50 Explain facility. The target maintains this information in a database that the origin may access via the Z39.50 Search and Present facilities.

This "explain" database appears to the origin as any other database supported by the target, but it has a well-known name and a pre-defined record syntax. Also, certain search terms, corresponding to information categories, are predefined in order to allow a semantic level of interoperability. Each information category has its own record layout, and all are included in the Explain syntax.

3.2 Facilities of the Information Retrieval Service

Sections 3.2.1 through 3.2.11 describe the eleven facilities of the Information Retrieval service. Most consist of logical groups of services; in several cases, a facility consists of a single service. Additional services may be added to any facility in future versions of this standard. Following is a summary description of the eleven facilities.

Initialization Facility--Init Service: A confirmed service initiated by the origin to initiate an Init operation.

Search Facility--Search Service: A confirmed service initiated by the origin to initiate a Search operation.

Retrieval Facility--The Retrieval facility consists of two services:

- Present Service: A confirmed service initiated by the origin to initiate a Present operation.
- Segment Service: A non-confirmed service initiated by the target, during a Present operation. *Note:* a Present operation thus consists of a Present request followed by zero or more Segment requests followed by a Present response.

Result-set-delete Facility--Delete Service: A confirmed service initiated by the origin to initiate a Delete operation.

Browse Facility--Scan Service: A confirmed service initiated by the origin to initiate a Scan operation.

Sort Facility--Sort Service: A confirmed service initiated by the origin to initiate a Sort operation.

Access Control Facility--Access-control service: a confirmed service initiated by the target. It does not initiate an operation, and it might or might not be part of an active operation.

Accounting/Resource Control Facility--The Accounting/ Resource Control facility consists of three services:

- Resource-control Service: A conditionally-confirmed service initiated by the target. It does not initiate an operation, and it might or might not be part of an active operation.
- Trigger-resource-control Service: A non-confirmed service initiated by the origin during an operation.
- Resource-report Service: A confirmed service initiated by the origin to initiate a Resource-report operation.

Explain Facility--The Explain facility does not include any services, but uses the services of the Search and Retrieval facilities.

Extended Services Facility--Extended-services Service: A confirmed service initiated by the origin to initiate an Extended-services operation.

Termination Facility--Close Service: A confirmed service initiated by the origin or target. It does not initiate nor is it part of any operation. It allows an origin or target to abruptly terminate all active operations and to initiate termination of the Z-Association. (Following termination of the Z-Association the origin may subsequently attempt to initialize another Z-Association using the Init service.)

3.2.1 Initialization Facility

The Initialization facility consists of the single service, Init.

3.2.1.1 Init Service

The Init service allows the origin to establish a Z-association. In the Init request, the origin proposes values for initialization parameters. In the Init response, the target responds with values for the initialization parameters; those values, which may differ from the origin-proposed values, are in effect for the Z-association.

If the target responds affirmatively (Result = 'accept'), the Z-association is established. If the origin then does not wish to accept the values in the target response, it may terminate the Z-association, via the Close service (and may subsequently attempt to initialize again). If the target responds negatively, the origin may attempt to initialize again.

Parameter	Origin Request	Target Response
Version	x	x
Id/authentication	x (opt)	
Options	x	x
Preferred-message-size	x	x
Exceptional-record-size	x	x
Result		x
Implementation-id	x (opt)	x (opt)
Implementation-name	x (opt)	x (opt)
Implementation-version	x (opt)	x (opt)
User-information-field	x (opt)	x (opt)
Other-information	x (opt)	x (opt)
Reference-id	x (opt)	x (if appl)

3.2.1.1.1 Version Both the origin and target indicate all versions that they support. The highest common version is selected for use, and is said to be 'in force,' for the Z-association. If there are no versions in common, the target should indicate 'reject' for the parameter Result.

Notes:

1. Version numbers higher than the highest known version should be ignored.
2. 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).

3.2.1.1.2 Id/authentication The origin and target agree, outside the scope of the standard, whether or not this parameter is to be supplied by the origin, and if so, to the value. This value is used by the target to determine if the origin is authorized to enter into communication with the target.

3.2.1.1.3 Options For each of the capabilities listed below, the origin proposes either ’on’ or ’off’ (meaning ’in effect’ or ’not in effect’ respectively) and the target responds correspondingly for each. The response determines whether the capability is in effect.

- search
- present
- delete
- resource-report
- scan
- sort
- extended-services
- trigger-resource-control
- level 1 segmentation
- level 2 segmentation
- concurrent operations
- named result sets
- resource-control
- access-control.

Note: the above list of capabilities is subject to expansion in future versions of this protocol.

The following rules, describing how these capabilities are to be negotiated, are intended to allow interoperation even when the origin and target have not necessarily implemented the same capabilities.

The Options parameter consists of a string of boolean flags, each corresponding to an individual capability. The origin might set the flag to ’in effect’ for a capability unknown to the target. In that case it is recommended that the target set the corresponding flag to ’not in effect’ in the response. However, if the origin sets a flag to ’not in effect’ and the target sets the corresponding flag to ’in effect,’ and if the origin is not aware what capability that flag represents, it is recommended that the origin terminate the Z-association.

search, present, delete, resource-report, scan, sort, and extended-services -- for each of these operation types, the origin indicates whether it wishes to initiate operations of that type; if so, the target indicates whether it is willing to process an operation of that type. If the origin proposes ’not in effect’ for a particular operation type, the target must also specify ’not in effect’.

Notes:

1. The target indication that it is willing to process a Resource-report operation does not imply that it will include a resource report in the response.
2. Any of the above operation types may be negotiated for any version. In particular, Scan, Sort, and Extended-services may be negotiated when version 2 is in force, even though they are not defined in ANSI Z39.50-1992.

trigger-resource-control -- The origin may propose to submit Trigger-resource-control requests; if so, the target indicates whether it will accept Trigger-resource-control requests. If the origin proposes ’not in effect,’ the target must also specify ’not in effect’.

Notes:

1. If the target specifies ’in effect’ for Trigger-resource-control, but ’not in effect’ for ’resource-control,’ then the origin may use only the Cancel function of Trigger-resource-control.
2. The target may indicate unwillingness to accept Trigger-resource-control requests even if it specifies ’in effect’ for ’resource-control’.
3. The target’s indication of willingness to accept Trigger-resource-control requests does not imply that the target will take any action as a result of a Trigger-resource-control request.

level 1 and level 2 segmentation -- The origin proposes one of the following:

- "no segmentation", by specifying ’not in effect’ for both level 1 and level 2 segmentation;
- "level 1 segmentation", by specifying ’in effect’ for level 1 and ’not in effect’ for level 2 segmentation; or
- "level 2 segmentation", by specifying ’in effect’ for level 2 segmentation.

Notes:

1. If the origin proposes ’in effect’ for level 2 segmentation then it may also propose ’in effect’ for level 1 segmentation to indicate that if the target is unable to support level 2 segmentation, the origin wishes level 1 segmentation to be in effect.

2. "segmentation" is said to be ’in effect’ if either level 1 or level 2 segmentation is in effect.
3. Segmentation may be in effect only when version 3 is in force.

The target response indicates which, if either, form of segmentation it intends to perform.

- If the target specifies neither level 1 nor level 2 then ’no segmentation’ is in effect, regardless or what the origin has proposed.
- If the target specifies level 1 (but not level 2) segmentation, it will not perform level 2 segmentation, and the origin must be prepared to accept level 1 segmentation, regardless of what the origin has proposed.
- If the target specifies level 2 segmentation, the origin must be prepared to accept level 2 segmentation regardless of what it has proposed (the target value for level 1 should be ’not in effect’).

When ’no segmentation’ is in effect, the target response to a Present request must consist of a single message (a single "segment", i.e. a Present response only, with no intervening Segment requests), containing an integral number of records. When ’Level 1 segmentation’ is in effect the target may respond to a Present request with multiple segments (i.e. a Present response, with possibly one or more intervening Segment requests); each must contain an integral number of records. When ’Level 2 segmentation’ is in effect the target may respond to a Present request with multiple segments, and individual records may span segments. Segmentation procedures are detailed in 3.3.

concurrent-operations -- The origin may propose to initiate concurrent operations; if so, the target indicates whether it will accept concurrent operations. If the origin proposes ’not in effect,’ the target must also specify ’not in effect’. If concurrent operations is not in effect, then ’serial operations’ is said to be in effect. Concurrent operations may be in effect only when version 3 is in force.

named-result-sets -- The origin may propose to use named-result sets (i.e. to specify result set names other than "default" as the value of Result-set-name within a Search request); if so, the target specifies whether it will support named-result-sets. If the origin proposes ’not in effect,’ the target must also specify ’not in effect’.

resource-control and access-control -- The origin indicates whether it wishes the target to invoke Resource-control and/or Access-control (i.e. send Resource-control and/or Access-control requests). The target specifies whether it plans to invoke Resource-control and/or Access-control.

Notes:

1. If the target specifies ’not in effect’ for resource-control (or access-control) then it will not invoke resource-control (or access control) even if the origin has proposed ’in effect’.
2. If the origin proposes ’not in effect’ for resource-control, and the target indicates ’in effect’ for resource-control, indicating that it is not willing to suppress Resource-control requests, and if indeed the origin cannot accept Resource-control requests, the origin should terminate the Z-association.
3. If the origin proposes ’not in effect’ for access-control, and if security requirements on the target system mandate that security (other than that which might be provided by the parameter Id/-authentication) be invoked at the outset of a Z-association, then the target should reject the Z-association (by setting the parameter Result to ’reject,’ and specifying ’in effect’ for ’access-control’).

   However, security may be invoked at different levels. In addition to authentication at the outset of a Z-association, security might be invoked to control access to a particular database, record, result-set, resource-report format, or use of an operation. Thus if the origin proposes ’not in effect’ for access-control, and the target normally invokes security (other than at the Z-association level), the target need not necessarily reject the Z-association.

   The target might wish to invoke a security challenge during an Init operation to determine whether the origin is authorized to use a capability it has proposed. If the origin has proposed ’not in effect’ for access-control, the target may simply refuse the use of that particular operation via the Options parameter.

   If the origin proposes ’not in effect’ for access-control, and the target chooses to accept the Z-association, and if the origin subsequently initiates an action that would precipitate an Access-control request (for example, if the origin issues a Search specifying a database for which it has not yet established credentials), the target should suppress the Access-control request and instead respond with an error status indicating that a security challenge was required but could not be issued.

3.2.1.1.4 Preferred-message-size and Exceptional-record-size The Init request contains the origin's proposed values of Preferred-message-size and Exceptional-record-size, specified in bytes. The Init response contains the Preferred-message-size and Exceptional-record-size that the target is going to use; these may be different from (and override) the values proposed by the origin. For both the request and response, Preferred-message-size must be less than or equal to Exceptional-record-size.

Exceptional-record-size is meaningful during a Present operation, and only in the special case when a single, exceptionally large record (i.e. larger than preferred-message-size) is requested in the Present request. In this special case, preferred-message-size may be overridden (for the present operation), so that a single record may be presented whose size may be as large as Exceptional-record-size. The fact that a single record is requested is how the origin signals that preferred-message-size may be overridden. Thus Exceptional-record-size must be greater than or equal to preferred-message-size. In the case where they are equal, Exceptional-record-size has no meaning (this is the way to signify that the special case will not apply during the Z-association).

The usage of these parameters is detailed in 3.3.

Note: The parameter Exceptional-record-size has the same meaning as the parameter Maximum-record-size defined in Z39.50-1992. The name of the parameter has been changed, for clarity.

3.2.1.1.5 Result The target indicates whether or not it accepts the Z-association by specifying a value of 'accept' or 'reject' in the parameter Result. (If 'reject' is indicated, the origin may send another Init request.)

3.2.1.1.6 Implementation-id, Implementation-name, and Implementation-version The request or response may optionally include any of these three parameters. They are, respectively, an identifier (unique within the client or server system), descriptive name, and descriptive version, for the origin or target implementation. These three implementation parameters are provided solely for the convenience of implementors, for the purpose of distinguishing implementations.

3.2.1.1.7 User-information-field This parameter may be used by the origin or target for additional information not specified by this standard.

3.2.1.1.8 Other-information This parameter may be used by the origin or target for additional information not specified by the standard. This parameter may be used only if version 3 is in force.

Note: Care should be taken by the origin when using this parameter; the origin cannot ascertain that version 3 is in force before sending the Init request.

3.2.1.1.9 Reference-id See 3.4.

3.2.2 Search Facility

The Search facility consists of the single service, Search.

3.2.2.1 Search Service

The Search service enables an origin to query databases at a target system, and to receive information about the results of the query.

The search request allows the origin to request that the target apply a query to a specified set of databases at the target, to identify records with the properties indicated by the query. The target creates a result set, which represents the set of records identified by the query, and the target maintains the result set for subsequent retrieval requests.

Depending on the parameters of the search, one or more records identified by the result set may be immediately retrieved as part of the search response. The result set is an ordered set; a record identified by an entry in the result set is referenced by the position of the entry within the result set (beginning with 1).

Parameter	Origin Request	Target Response
Query-type	x	
Query	x	
Database-names	x	
Result-set-name	x	
Replace-indicator	x	
Small-set-element-set-names	x (opt.)	
Medium-set-element-set-names	x (opt.)	
Preferred-record-syntax	x (opt.)	
Small-set-upper-bound	x	
Large-set-lower-bound	x	
Medium-set-present-number	x	
Response-records		x (if appl.)
Result-count		x
Number-of-records-returned		x
Next-result-set-position		x
Search-status		x
Result-set-status		x (if appl.)
Present-status		x (if appl.)
Additional-search-information	x (opt.)	x (opt.)
Other-information	x (opt.)	x (opt.)
Reference-id	x (opt.)	x (if appl.)

3.2.2.1.1 Query-type and Query The parameter Query-type identifies the type of query, i.e the syntax of parameter Query. Six types are defined:

- Type-0 may be used only when the origin and target have a priori agreement outside of the standard.
- Type-1 is the Reverse Polish Notation (RPN) query specified in 3.7.
- Type-2 is the ISO8777 type query, specified in ISO 8777.
- type-100 is the Z39.58 type query, specified in ANSI Z39.58.
- Type-101 is the extended RPN (ERPN) query, an extension to the type-1 query to allow proximity searching and restriction of result sets by attributes. It is specified in 3.7.
  Note: The type-101 query is identical to the type-1 query with the following exception: For type-1, proximity and restriction are valid only when version 3 is in force. For type-101, proximity and restriction are valid both for version 3 and version 2 as well. (The definition of the type-101 query is independent of version.)
- Type-102 is the Ranked List query, to be defined in a later version of this standard.

3.2.2.1.2 Database-names The origin indicates the set of databases to which the Query applies.

Notes:

1. The target designates (through the Explain facility or through some mechanism outside of the standard) what databases may be specified on a Search request, and in what combinations they may be specified. For example, a target might specify that databases **A**, **B**, and **C**, may be searched individually, and that **A** and **B** may be searched in combination (but not **A** and **C**, nor **B** and **C**).
2. Each database name specified by the target is a string of characters, and the string is case-insensitive. That is, for any character that is a letter, the origin may use either upper or lower case, regardless of how the target has specified the name.

3.2.2.1.3 Result-set-name and Replace-indicator The parameter Result-set-name specifies a name to be given to the result set (to be created by the query) so that it may be subsequently referenced (within the same Z-association).

If a result set with the same name already exists at the target, the action taken depends on the value of the parameter Replace-indicator, as follows:

- If the value of Replace-indicator is 'on,' then after processing the query, the existing result set whose name is specified by the parameter Result-set-name will be deleted, and a new result set by that name created. If the search cannot be processed, the content of the result set will be empty.
- If the value of Replace-indicator is 'off,' the search is not processed, an error diagnostic is returned by the target, and the existing result set whose name is specified by the parameter Result-set-name is left unchanged.

If a result set does not exist with the name specified by the parameter Result-set-name, then a result set by that name is created by the target, and the parameter Replace-indicator is ignored. The initial content of the result set is empty. If no records are found by the query, the result set remains empty.

A target need not support, in general, the naming of result sets by the origin. However, a target must support at least the result set whose name is "default." If the origin specifies "default" as Result-set-name, then Replace-indicator must be 'on'.

A result set created by a Search request (that is, specified by the parameter Result-set-name) may be referenced in a subsequent Present request or as an operand in a subsequent Search request (for example, in a type-1 query). If a result set named "default" is created, it remains available for reference from the time it is created until the end of the Z-association during which it is created, or until either:

- another default result set is created, because the name "default" is specified as Result-set-name in a subsequent Search request, or
- it is unilaterally erased or deleted by the target.

Any result set other than the result set named "default" remains available for reference from the time it is created until it is deleted in one of the following ways:

- by a Delete operation
- implicitly, because a result set was specified by the same name in a Search request, and the value of the parameter Replace-indicator was 'on'
- unilaterally by the target (at any time)
- by termination of the Z-association.

3.2.2.1.4 Small-set-element-set-names and Medium-set-element-set-names These parameters describe the preferred composition of the records expected in the search response. If the query results in a small-set (see 3.2.2.1.6), then Small-set-element-set-names pertains. If the query results in a medium-set, then Medium-set-element-set-names pertains. These two parameters are described in 3.6.2.

3.2.2.1.5 Preferred-record-syntax The origin may specify a preferred record syntax for retrieval records. If the target cannot supply a particular record according to the Preferred-record-syntax, it supplies the record according to one of the other abstract syntaxes from the set for which Presentation contexts are currently established for this A-association.

If the target cannot supply the record according to either the requested syntax or a syntax corresponding to an established presentation context, it returns a surrogate diagnostic for that record, unless the established set of presentation contexts is empty; in that case, this standard does not prescribe the target action.

3.2.2.1.6 Small-set-upper-bound, Large-set-lower-bound, and Medium-set-present-number The result set is considered a "small-set," "medium-set," or "large-set," depending on the values of parameters Small-set-upper-bound and Large-set-lower-bound of the Search request, and Result-count of the Search response (see 3.2.2.1.8). The result set is a small-set if Result-count is not greater than small-set-upper-bound. The result set is a large-set if Result-count is larger than or equal to Large-set-lower-bound. Otherwise, the result set is a medium-set. If the query results in a small-set, response records corresponding to all database records identified by the result set are to be returned to the origin (subject to possible message size constraints). If the query results in a large-set, no response records are to be returned. If the query results in a medium-set, the maximum number of response records to be returned is specified by Medium-set-present-number.

Notes:

1. The result set may be a medium-set only when Result-count is greater than small-set-upper-bound and less than Large-set-lower-bound, and this can occur only if Large-set-lower-bound is at least 2 greater than Small-set-upper-bound; i.e., the result set cannot be a medium-set if Large-set-lower-bound exceeds Small-set-upper-bound by 1. For example, if Large-set-lower-bound is 11 and Small-set-upper-bound is 10, the intent is "if 10 or less database records are found, return response records for them all, otherwise do not return any," and medium-set-present-number would not apply.
2. Small-set-upper-bound may be zero. Large-set-lower-bound must be greater than Small-set-upper-bound.
3. If the origin does not want any response records returned regardless of the value of Result-count, Large-set-lower-bound should be set to 1 and Small-set-upper-bound to zero.

3.2.2.1.7 Response-records The target processes the search, creating a result set that identifies a set of database records. It cannot be assumed however that search processing requires physical access to the database records. A particular database record might not be accessible but this circumstance might not be recognized until an attempt is made to access the record for the purpose of forming a retrieval record.

After processing the search, the target attempts to create retrieval records to be included in the Search response, corresponding to the first N database records identified by the result set (N depends on the request parameters and Result-count, as described in 3.2.2.1.6). For each database record for which a retrieval record cannot be included, a surrogate diagnostic record is substituted.

The term response record refers to a retrieval record or a surrogate diagnostic record. The parameter Response-records is one of the following:

- N response records;
- a number of response records, which is less than N because of message size constraints (see 3.3);

- one or more non-surrogate diagnostic records (see note) indicating that the search cannot be processed, and why it cannot be processed; or
- one or more non-surrogate diagnostic records (see note) indicating that records cannot be presented, and why not, e.g. "element set name not valid for database."

  Note: If version 2 is in force, the target returns a single non-surrogate diagnostic record. If version 3 is in force, the target returns one or more non-surrogate diagnostic records.

The order of occurrence of response records within the parameter Response-records is according to the order in which they are identified by the result set. Each may optionally be accompanied by the name of the database in which the record resides. However, the database name must accompany the first response record being returned, and must accompany any record from a database different from its immediate predecessor.

3.2.2.1.8 Result-count and Number-of-records-returned The parameter Result-count is the number of database records identified by the result set. If the result set is empty, result-count is zero. The parameter Number- of-records-returned is the total number of records returned in the Search response.

3.2.2.1.9 Next-result-set-position The parameter Next-result-set-position takes on the value M+1, where M is the position of the result set item which identifies the database record corresponding to the last response record among those returned; or zero if M = Result-count.

3.2.2.1.10 Search-status The parameter Search-status, returned in the response, assumes one of the following two values:

success	-	The search completed successfully.
failure	-	The search did not complete successfully.

A value of ’success’ does not imply that the expected response records are being returned as part of the response (see Present-status, 3.2.2.1.11). Note also, a value of ’success’ does not imply that any database records were located by the search. A value of ’failure’ *does* imply that *none* of the expected response records is being returned. In the latter case, the target returns one or more non-surrogate diagnostic records (see note) indicating that the search cannot be processed.

Note: If version 2 is in force, the target returns a single non-surrogate diagnostic record. If version 3 is in force, the target returns one or more non-surrogate diagnostic records.

3.2.2.1.11 Result-set-status and Present-status These are status descriptors necessary to distinguish potentially ambiguous situations that can occur during search and present operations.

Result-set-status occurs if and only if the value of Search-status is ’failure,’ and its value is one of the following:

subset	-	Partial, valid results available.
interim	-	Partial results available, not necessarily valid.
none	-	No result set.

Present-status occurs if and only if the value of Search-status is ’success,’ and its value is one of the following:

success	-	All of the expected response records are available.
partial-1	-	Not all of the expected response records can be returned because the request was terminated by access control.
partial-2	-	Not all of the expected response records can be returned because they will not fit within the preferred message size.
partial-3	-	Not all of the expected response records can be returned because the request was terminated by resource control, at origin request.
partial-4	-	Not all of the expected response records can be returned because the request was terminated by resource control, by the target.
failure	-	None of the expected response records can be returned. One or more non-surrogate diagnostic records is returned (see note in 3.2.2.1.7).

3.2.2.1.12 Additional-search-information On the response the target may use this parameter to convey information which is a by-product of the search process, including, for example, intermediate result counts, why particular records were returned, or whether a particular attribute was used for searching a database. On the request the origin may use this parameter to indicate the preferred format or content of that information. User Information format SearchResponse-1 is defined in Appendix 8 USR. This parameter may be used only when version 3 is in force.

3.2.2.1.13 Other-information This parameter may be used by either the origin or target for additional information not specified by the standard. This parameter may be used only when version 3 is in force.

3.2.2.1.14 Reference-id See 3.4.

3.2.3 Retrieval Facility

The Retrieval facility consists of two services: Present and Segment.

The origin sends a Present request to request response records according to position within a result set maintained by the target. The target responds by sending a Present response, containing the requested response records. Alternatively, if segmentation is in effect and the requested response records will not fit within the Present response message, the target may segment the response by sending one or more Segment requests before the Present response. The procedures for segmentation are described in 3.3.

The Segment requests (if any) together with the Present response are referred to as the aggregate Present response. Each Segment request as well as the Present response is referred to as a segment of the Present response. If an aggregate Present response consists of a single segment (i.e. only a Present response) it is called a simple Present response.

3.2.3.1 Present Service

The Present service allows the origin to request response records corresponding to database records represented by a specified result set. Database records are referenced by relative position within the result set. The origin specifies a range and may follow with subsequent requests specifying different ranges.

Note: If version 3 is in force, a single request may include more than one range.

The origin may request, for example, records one through five and follow with a request for records four through six.

Note: In this section, "record N" means "the response record corresponding to the database record identified by result set entry N."

Parameter	**Origin Request**	**Target Response**
Number-of-records-requested	x	
Result-set-start-position	x	
Additional-ranges	x (opt.)	
Result-set-id	x	
Element-set-names	x (opt.)	
Preferred-record-syntax	x (opt.)	
Comp-spec	x (opt.)	
Max-segment-count	x (opt.)	
Max-segment-size	x (opt.)	
Max-record-size	x (opt.)	
Response-records		x (if appl.)
Number-of-records-returned		x
Next-result-set-position		x
Present-status		x
Other-information	x (opt.)	x (opt.)
Reference-id	x (opt.)	x (if appl.)

3.2.3.1.1 Number-of-records-requested and Result-set-start-position The origin requests a range of records: N records beginning at record M. M = Result-set-start-position, N = Number-of-records-requested and N is not greater than (Result-count - M) + 1.

3.2.3.1.2 Additional-ranges The origin may request additional ranges of records by including this parameter, which consists of one or more pairs (M, N) where M and N are as described in 3.2.3.1.1. For the first pair (M, N) M must be greater than or equal the sum of Result-set-start-position and Number-of-records-requested. For any consecutive pairs (M1, N1) and (M2, N2), M1 + N1 must be less than M2. This parameter may occur only when version 3 is in force.

3.2.3.1.3 Result-set-id The origin indicates the name of a transient result set, created during this Z-association, from which records are to be retrieved.

3.2.3.1.4 Element-set-names The origin may indicate the desired composition of the retrieved records. See 3.6.2.

3.2.3.1.5 Preferred-record-syntax See 3.2.2.1.5.

3.2.3.1.6 Comp-spec This parameter may be included only if the parameter Element-set-names is omitted, and only if version 3 is in force. If included, Comp-spec provides an alternative means of specifying the desired composition of retrieved records. See 3.6.

3.2.3.1.7 Max-segment-count, Max-segment-size, and Max-record-size These three parameters may be used only when version 3 is in force.

Max-segment-count may be included when level-1 or level-2 segmentation is in effect; it specifies the maximum number of segments the target may include in the aggregate Present response. If its value is 1, no segmentation is applied for the operation and Max-record-size should not be included.

Max-segment-size and/or Max-record-size may be included only when level 2 segmentation is in effect. Max-segment-size is the largest allowable segment; if included, it overrides Preferred-message-size (for this Present operation only); if not included it assumes the value of Preferred-message-size. Max-record-size is the largest allowable retrieval record within the aggregate Present response; if included, it must equal or exceed Max-segment-size.

These three parameters are further detailed in 3.3.3.2.

3.2.3.1.8 Response-records This parameter consists of a sequence of response records, or possibly, if 'level 2 segmentation' is in effect, a final fragment (see 3.3.3) followed by zero or more response records. Alternatively (if the operation included no Segment requests) the parameter consists of one or more non-surrogate diagnostic records indicating that the request cannot be processed, and why not (see note below).

A response record will be returned in the aggregate Present response for each record requested in the request (subject to message size, access-control, and resource-control constraints). Each response record corresponds to a result set entry, and the result set ordinal positions represented by the response records will be ascending and consecutive, unless the request included the parameter Additional-ranges. In this case, the positions will be ascending but may have gaps which will correspond exactly to the gaps in the requested ranges.

Each response record may optionally be accompanied by the name of the database to which it corresponds. However, the database name must accompany the first response record (or starting fragment) within the first segment of the aggregate Present response, and must accompany any response record (or starting fragment of a response record) from a database different from its immediate predecessor within the aggregate Present response.

When the origin has received the aggregate Present response, the result (if all of the segments are re-assembled, and segmented response records re-assembled from their fragments) will be one of the following:

- N response records, where N = Number-of-records-requested,
- a number of response records, which is less than N (reason specified by Present-status), or
- one or more diagnostic records (see note) indicating that the request cannot be processed, and why not.

Note: If version 2 is in force, the target returns a single non-surrogate diagnostic record. If version 3 is in force, the target returns one or more non-surrogate diagnostic records.

3.2.3.1.9 Number-of-records-returned and Next-result-set-position The parameter Number-of-records-returned is the total number of records in the aggregate Present response. Next-result-set-position is the value M+1, where M is the position of the result set item corresponding to the last record among those included in the response; or zero if M is the position of the last result set item.

3.2.3.1.10 Present-status Present-status is mandatory in a Present response and its values are the same as those listed for Present-status in 3.2.2.1.11. Present-status refers to the aggregate Present response.

3.2.3.1.11 Other-information This parameter may be used by the origin or target for additional information, not specified by the standard. This parameter may be used only if version 3 is in force.

3.2.3.1.12 Reference-id See 3.4.

3.2.3.2 Segment Service

If the records requested by a Present request will not fit in a single segment, and if segmentation is in effect, the target returns multiple segments, each of which contains a portion of the records. Each except the last segment is returned as a Segment request (the last segment is returned as a Present response).

Notes:

1. The segment service is modelled as a request, even though, logically, the target is not making a request. The reason is that (for purposes of abstract service definition and resultant protocol specification) any message is a request or a response, a response must be preceded by a request of the same type, and there may be at most one response to a given request. Because of these

modelling constraints: the Segment service cannot be modelled as a response (because if it were, it would necessarily respond to a segment request, and it is a non-confirmed service); and the present operation cannot be modelled as a Present request followed by multiple Present responses.
2. This service may be used only when version 3 is in force.
3. If segmentation is not in effect, the target does not send any Segment requests and the aggregate Present response consists of a simple Present response. If the records requested will not fit in a segment, the procedures described in 3.3.1 apply.
4. If the records requested will fit in a single segment (whether or not segmentation is in effect) the target does not send any Segment requests and the aggregate Present response consists of a simple Present response.

Parameter	Target Request
Segment-records	x
Number-of-records-returned	x
Other-information	x (optional)
Reference-id	x (if applicable)

3.2.3.2.1 Segment-records If level 1 segmentation is in effect, the parameter Segment-records consists of a sequence of response records.

If level 2 segmentation is in effect, the parameter Segment-records may include response records as well as fragments (see 3.3.3). It may be composed of a final fragment (except within the first segment of the aggregate Present response), followed by zero or more response records, followed by a starting fragment. Neither fragment need occur, however if neither occurs there must be at least one response record. (Note that fragments pertain only to retrieval records; a diagnostic record may not be segmented.)

The order of occurrence of a response record or fragment of a retrieval record is according to the order in which the record is identified by the result set. Each response record or starting fragment may optionally be accompanied by the name of the database to which it pertains. However, the database name must accompany the first response record (or starting fragment) within the first segment of the aggregate Present response, and must accompany any response record (or starting fragment of a retrieval record) from a database different from its immediate predecessor within the aggregate Present response.

3.2.3.2.2 Number-of-records-returned This is the total number of response records and starting fragments included within the segment.

3.2.3.2.3 Other-information This parameter may be used by the target for additional information, not specified by the standard.

3.2.3.2.4 Reference-id See 3.4.

3.2.4 Result-set-delete Facility

The Result-set-delete facility consists of a single service, Delete.

3.2.4.1 Delete Service

The Delete service enables an origin to request that the target delete specified result sets, or all result sets, created during the Z-association. The target responds by reporting information pertaining to the result of the operation.

Parameter	Origin Request	Target Response
Delete-function	x	
Result-set-list	x (if appl.)	
Delete-operation-status		x
Delete-list-statuses		x (if appl.)
Number-not-deleted		x (if appl.)
Bulk-statuses		x (if appl.)
Delete-msg		x (opt.)
Other-information	x (opt.)	x (opt.)
Reference-id	x (opt.)	x (if appl.)

3.2.4.1.1 Delete-function The origin specifies one of the following:

list	-	delete specified result sets (see 3.2.4.1.2), or
bulk-delete	-	delete all result sets currently on the target created during this Z-association.

3.2.4.1.2 Result-set-list This parameter occurs if and only if Delete-function is 'list'. It contains a list of result sets (created during this Z-association) to be deleted.

3.2.4.1.3 Delete-operation-status Delete-operation-status is the status of the delete request. It assumes one of the values 'success' or 'failure-3' through 'failure-9' in the table below.

3.2.4.1.4 Delete-list-statuses Delete-list-statuses is present in a Delete response if Delete-function in the request was 'list'. Delete-list-statuses contains the same list of result sets as in the Result-set-list parameter of the Delete request, each paired with a status. Possible status values are 'success,' 'failure-1' through 'failure-6,' and 'failure-10'.

Status	Description
success	Result set(s) deleted.
failure-1	Result set did not exist.
failure-2	Result set previously unilaterally deleted by target.
failure-3	System problem at target (optional text message may be included in the Delete-msg parameter).
failure-4	Access-control failure: the delete request caused the target to issue an Access-control request which the origin failed to satisfy, or the origin could not accept an Access-control request.
failure-5	Operation terminated by resource control at origin request.
failure-6	Operation terminated by target due to resource constraints.
failure-7	Bulk delete of result sets not supported by target.
failure-8	Not all result sets deleted (on a bulk-delete request) (see 3.2.4.1.5).
failure-9	Not all requested result sets deleted (on a list request).
failure-10	Result-set in use.

Notes:

1. failure-7 and failure-8 can occur only if Delete-operation is Bulk-delete.
2. failure-10 may be used only when version 3 is in force.

3.2.4.1.5 Number-not-deleted and Bulk-statuses These two parameters occur only if Delete-function is Bulk-delete and if Delete-operation-status = 'failure-8'. The parameter Number-not-deleted indicates how many result sets were not deleted, and the parameter Bulk-statuses gives individual statuses for those not deleted.

Note, however, that a target is not obligated to provide statuses for each result set not deleted on a bulk delete. For example, a target may abort a bulk delete when the first failure to delete a result set that is part of the bulk delete fails; in this case only a single status might be included in the parameter Bulk-statuses.

If a bulk delete results in more statuses than can fit into a single Delete-response message, the target may discard those that do not fit.

3.2.4.1.6 Delete-msg Delete-msg, if present, contains an optional text message.

3.2.4.1.7 Other-information This parameter may be used by either the origin or target for additional information not specified by the standard. This parameter may be used only when version 3 is in force.

3.2.4.1.8 Reference-id See 3.4.

3.2.5 Access Control Facility

The Access Control facility consists of a single service, Access-control.

3.2.5.1 Access-control Service

The Access-control service allows a target to challenge an origin. The challenge might pertain to a specific operation or to the Z-association. The Access-control request/response mechanism can be used to support access control challenges or authentication, including password challenges, public key cryptosystems, and algorithmic authentication.

An origin must be prepared to accept and respond to Access-control requests from the target if access control is in effect. A target may issue an Access-control request which is either part of a specific (active) operation, or which pertains to the Z-association.

- If concurrent operations is in effect:
  - -- If the Access-control request includes a Reference-id: The supplied Reference-id must correspond to an active operation; the Access-control request is part of that operation. The Access-control response must also include that Reference-id.
  - -- If the Access-control request does not include a Reference-id: The Access-control request and response are not part of any operation, they pertain to the Z-association.
- If serial operations is in effect: The target may issue an Access-control request only when there is an active operation; the Access-control request and subsequent response are part of that operation and must include the Reference-id of the

operation (which is assumed 'null' if not present in the initiating request).

The following procedures pertain to access control as it applies to an operation:

1. After sending an initiating request, the origin must be prepared to receive an Access-control request (for that operation), respond with an Access-control response, then later receive another Access-control request, etc., before receiving a terminating response. The target might suspend processing of the operation from the time that it sends the Access-control request until it receives the Access-control response. The challenge does not interrupt any other operation. If the origin response is acceptable to the target, the operation proceeds as if the challenge has never taken place. If the origin fails to respond correctly to the challenge then the target's terminating response to the interrupted operation may indicate that the operation was terminated due to an Access-control failure.
2. If the origin fails to respond correctly to a challenge during an Init operation, the target may reject the Z-association (by setting the Result parameter to 'reject,' and optionally supplying an explanatory message in the User-information-field of the Init response). However, the target need not necessarily reject the Z-association. For example the target might wish to invoke a security challenge during an Init operation to determine whether the origin is authorized to use a capability it has proposed. If the origin fails to respond properly, the target may simply refuse the use of that particular operation (via the Options parameter).
3. During a Search or Present operation, while the target is preparing records for presentation, it might send an Access-control request pertaining to a particular record. If the origin fails to respond correctly to the challenge, the target may simply substitute a surrogate diagnostic: "security challenge failed; record not included."

The following procedures pertain to access control as it applies to the Z-association:

1. If concurrent operations is in effect, following initialization the origin must be prepared at any time during the association, whether or not operations are active, to receive an Access-control request pertaining to the Z-association, to respond with an Access-control response, then later to receive another Access-control request, etc.
2. The target might suspend processing of some or all of the active operations from the time that it sends the Access-control request until it receives the Access-control response. If the origin response is acceptable to the target, the suspended operations proceed as if the challenge had never taken place.
3. If the origin fails to respond correctly to the challenge, the target might decide to terminate one or more operations but to leave open the Z-association. In that case, the target's terminating response to any such operations may indicate that the operation was terminated because of an Access control failure. Alternatively, the target may close the Z-association.

Parameter	**Target Request**	**Origin Response**
Security-challenge	x	
Security-challenge-response		x
Other-information	x (opt.)	x (opt.)
Reference-id	x (if appl.)	x (if appl.)

3.2.5.1.1 Security-challenge and Security-challenge-response Definitions for format and content of the challenge and response are subject to registration; several definitions are defined and registered in Appendix 7 ACC. Alternatively, the contents of these two parameters may be established by prior agreement between a given target/origin pair.

3.2.5.1.2 Other-information This parameter may be used by either the origin or target for additional information not specified by the standard. This parameter may be used only when version 3 is in force.

3.2.5.1.3 Reference-id If serial operations is in effect, or if concurrent operations is in effect and the challenge pertains to a particular operation, then the use of Reference-id is governed by section 3.4. If 'concurrent operations' is in effect and the challenge pertains to the Z-association, then the Reference-id is to be omitted from both the request and response.

3.2.6 Accounting/Resource Control Facility

The Accounting/Resource Control facility consists of three services:

- the Resource-control service, invoked by the target, either as part of an active operation (of any type) or pertaining to the Z-association;
- the Trigger-resource-control service, invoked by the origin as part of an active operation (of any type except Init), and
- the Resource-report service, invoked by the origin to initiate a Resource-report operation.

The Resource-control service permits the target to send a Resource-control request, which might include a resource report. The report might notify the origin that either actual or predicted resource consumption will exceed agreed upon limits (or limits built into the target), and request the origin's consent to continue an operation, via the Resource-control response. The target might, for example, inform the origin about the current status of a result set being generated on the target during a Search operation, and indicate information about the progress of the operation.

The Trigger-resource-control service permits the origin to request that the target initiate the Resource-control service, or cancel the operation.

The Resource-report service permits the origin to request that the target send a Resource-report pertaining to a completed operation or to the Z-association.

3.2.6.1 Resource-control Service

An origin must be prepared to accept and respond to Resource-control requests from the target if resource control is in effect. A target may issue a Resource-control request which is either part of a specific (active) operation or which pertains to the Z-association.

- If concurrent operations is in effect:
  - -- If the Resource-control request includes a Reference-id: The supplied Reference-id must correspond to an active operation; the Resource-control request is part of that operation. The Resource-control response (if any) must also include that Reference-id.
  - -- If the Resource-control request does not include a Reference-id: The Resource-control request and response are not part of any operation, they pertain to the Z-association.
- If serial operations is in effect: The target may issue a Resource-control request only when there is an active operation; the Resource-control request and (possible) subsequent response are part of that operation and must include the Reference-id of the operation (which is assumed 'null' if not present in the initiating request).

The Resource-control request indicates whether a response is required:

- If so, the origin must issue a Resource-control response. If the Resource-control request was part of an operation: the response is part of the same operation; the target awaits the Resource-control response, and subsequently issues a terminating response after processing of the operation is concluded.
- If not, the origin must not issue a Resource-control response. If the Resource-control request was part of an operation: the target subsequently issues the terminating response, after processing of the operation is concluded.

An origin should be prepared to receive, and (conditionally) respond to, multiple Resource-control requests as part of an operation (while the operation is active), or pertaining to the Z-association.

If the origin responds to a Resource-control request with a Resource-control response saying to terminate an operation, it can expect to receive a terminating response. This response might indicate that the operation was terminated at origin request. However, the response might alternatively indicate that the operation completed, since the operation at the target may continue to execute and subsequently complete before the Resource-control response reaches the target.

Parameter	Target Request	Origin Response
Resource-report	x (opt.)	
Partial-results-available	x (if appl.)	
Suspended-flag	x (if appl.)	
Response-required	x	
Triggered-request-flag	x (opt.)	
Continue-flag		x
Result-set-wanted		x (if appl.)
Other-information	x (opt.)	x (opt.)
Reference-id	x (if appl.)	x (if appl.)

3.2.6.1.1 Resource-report This parameter may be used to convey information about the current and estimated resource consumption at the server. The format of Resource-report resource-1 and resource-2 are defined in Appendix 6 RSC.

3.2.6.1.2 Partial-results-available The target indicates the status of the result set via the flag Partial-results-available, whose value is one of the following:

subset	-	Partial, valid results available.
interim	-	Partial results available, not necessarily valid.
none	-	No results available.

This parameter is meaningful only as part of a search operation. If its value is 'subset' or 'interim,' then the target will accept subsequent Present requests against the result set if the origin indicates (via the Continue-flag) that the operation is to be terminated and if the value of the parameter Result-set-wanted is ’on’.

If the value of Partial-results-available is ’none’ then the target need not accept subsequent Present requests in the event that the origin indicates (via the Continue-flag) that the operation is to be terminated.

Note that if the Suspended-flag is off, the partial results available situation may change because processing of the Search operation may continue. In all cases, the values of Search-status and Result-set-status in the Search response should be treated as the authoritative information.

3.2.6.1.3 Suspended-flag This parameter is valid only when the request pertains to an operation. The target indicates whether or not processing of the operation has been suspended pending the Resource-control response. This flag occurs if and only if the value of Response-required is ’yes’.

3.2.6.1.4 Response-required The target indicates whether or not a response (from the origin) to this request is required.

3.6.2.1.5 Triggered-request-flag This parameter is valid only when the request pertains to an operation. The target may optionally indicate whether or not this request resulted from a Trigger-resource-control request from the origin.

3.2.6.1.6 Continue-flag This parameter is valid only when the request pertains to an operation. The origin indicates to the target whether or not to continue processing the operation.

3.2.6.1.7 Result-set-wanted This flag is valid only:

- during a Search operation,
- when the value of Partial-results-available is ’subset’ or ’interim,’ and
- when the value of the parameter Continue-flag is ’do not continue’.

If the value of this flag is ’yes,’ the target will maintain the (possibly partial) result set for subsequent Present operations. If the value of the flag is ’no,’ the target may delete the result set. A result set status of ’none’ on the subsequent Search response indicates that the target has discarded the result set. In all cases, the values of Search-status and Result-set-status in the Search response describe the actual decisions made by the target and the way in which the search terminated.

3.2.6.1.8 Other-information This parameter may be used by either the origin or target for additional information, not specified by the standard. This parameter may be used only when version 3 is in force.

3.2.6.1.9 Reference-id See 3.4.

3.2.6.2 Trigger-resource-control Service

An origin may issue Trigger-resource-control requests during an operation (except during an Init operation), as part of that operation. It serves as a signal to the target that the origin wishes the target to:

a) simply send a Resource-report, i.e., issue a Resource-control request with Response-required ’off’;
b) invoke full resource control, i.e., issue a Resource-control request with Response-required ’on’; or
c) cancel the operation.

The target is not obliged to take any specific action upon receipt of a Trigger-resource-control request. For the purpose of procedure description, there is no response to the request; if the target wishes to issue a Resource-control request it does so unilaterally. (If the origin issues a Trigger-resource-control request and subsequently receives a Resource-control request as part of the same operation, the origin cannot necessarily determine whether the latter resulted from the Trigger-resource-control request. However, the target may include Triggered-request-flag in the Resource-control-request to so indicate.)

If the origin issues a Trigger-resource-control request saying to cancel the operation, and if the target honors the request, the origin can expect to receive a terminating response indicating that the operation was terminated at origin request.

Although an origin may issue a Trigger-resource-control request as part of an active operation, the target might receive the request after the operation terminates. In that case, the target will ignore the Trigger-resource-control request. Furthermore, the target might receive a Trigger-resource-control request after issuing a Resource-control request for the same operation, while awaiting a Resource-control response. In that case, again, the target should ignore the Trigger-resource-control request. (Note that in general, the target may ignore any Trigger-resource-control request.)

Parameter	Origin Request
Requested-action	x
Preferred-resource-report-format	x (if applicable)
Result-set-wanted	x (if applicable)
Other-information	x (optional)
Reference-id	x (if applicable)

3.2.6.2.1 Requested-action The origin indicates one of the following:

resource-report	-	Issue a Resource-control request and set Response-required to 'off'.
resource-control	-	Issue a Resource-control request and set Response-required to 'on'.
cancel	-	Terminate the operation.

3.2.6.2.2 Preferred-Resource-report-format The origin may indicate a resource report format that it prefers.

3.2.6.2.3 Result-set-wanted This flag is meaningful only for a Search operation, and when the requested action is 'cancel'. If the value of the flag is 'yes,' the origin requests that the target maintain the (possibly partial) result set for subsequent Present operations. See 3.2.6.1.7.

3.2.6.2.4 Other-information This parameter may be used by the origin for additional information, not specified by the standard. This parameter may be used only when version 3 is in force.

3.2.6.2.5 Reference-id See 3.4.

3.2.6.3 Resource-report Service

The Resource-report service allows an origin to request a Resource-report, pertaining to a specified, completed operation, or to the entire Z-association.
Note: The Resource-report service differs from the Trigger-resource-control service, in this respect: Trigger-resource-control is a non-confirmed service; there is a request, but no response. The request is part of, but does not initiate, an operation; it requests a report pertaining to that active operation. Resource-report, in contrast, is a *confirmed* service; there is a request and a response (the target is obliged to respond, although the target is not obliged to include a resource report in the response). The request and response initiate and terminate an operation respectively; the request identifies a particular *completed* operation and solicits a report pertaining to that operation (or it may solicit a report pertaining to the entire Z-association).

Parameter	Origin Request	Target Response
Preferred-resource-report-format	x (opt.)	
Op-id	x (opt.)	
Resource-report-status		x
Resource-report		x (opt.)
Other-information	x (opt.)	x (opt.)
Reference-id	x (opt.)	x (if appl.)

3.2.6.3.1 Preferred-resource-report-format The origin may indicate a resource report format that it prefers.

3.2.6.3.2 Op-id This parameter may be supplied by the origin to identify a completed operation for which the origin requests a resource report. This parameter may be used only when version 3 is in force.

- If Op-id is present, it consists of a Reference-id, and refers to the most recently completed operation that used that Reference-id.
  Notes:
  1. When an operation terminates, if the origin anticipates that it will subsequently issue a Resource-report request pertaining to that operation, it is the origin's responsibility to ensure that the Reference-id is not re-used before doing so.
  2. The origin may (but need not) use the same reference-id for the Resource-report operation as that specified in Op-id, and if so, Op-id will nevertheless pertain to a completed operation only. However, it is recommended that the origin not specify a value of Op-id equal to any reference-id being used by any active operation other than this Resource-report operation. If the origin does so, the target may (but need not) consider the request in error (see failure-6 of Resource-report-status).
  3. If the origin wants resource information about an *active* operation, it should not use the Resource-report service, but instead use the Trigger-resource-control service, as part of that operation. If the operation terminates before the target receives the Trigger-resource-control request, the origin will receive a terminating response and may then subsequently issue a Resource-report request pertaining to that (completed) operation.

- If Op-id is not present, the origin requests a resource report pertaining to the Z-association.

3.2.6.3.3 Resource-report-status The target supplies one of following status values:

success	-	A resource report is included (and in the preferred format, if the parameter Preferred-resource-report-format was included in the request).
partial	-	A resource report is included, but not in the preferred format (applies only if the parameter Preferred-resource-report-format was included in the request).
failure-1	-	Target unable to supply resource report.
failure-2	-	Operation terminated by target due to resource constraints.
failure-3	-	Access-control failure.
failure-4	-	Unspecified failure.
failure-5	-	There is no known operation with specified id.
failure-6	-	There is an active operation with specified id.

Note: Failure-5 and failure-6 apply only when version 3 is in force.

3.2.6.3.4 Resource-report See 3.2.6.1.1.

3.2.6.3.5 Other-information This parameter may be used by either the origin or target for additional information, not specified by the standard. This parameter may be used only when version 3 is in force.

3.2.6.3.6 Reference-id See 3.4.

3.2.7 Sort Facility

The Sort facility consists of a single service, Sort.

3.2.7.1 Sort Service

The Sort service allows an origin to request that the target sort a result set (or merge multiple result sets and then sort). The origin specifies a sequence of sort elements. The result set is to be ordered according to the specified sequence, and subsequent positional requests against the result set will be construed by the target to apply to the result set as so ordered.

Parameter	Origin Request	Target Response
Input-result-sets	x	
Sorted-result-set	x	
Sort-sequence	x	
Sort-status		x
Result-set-status		x (if appl.)
Diagnostics		x (if appl.)
Other-information	x (opt.)	x (opt.)
Reference-id	x (opt.)	x (if appl.)

3.2.7.1.1 Input-result-sets This parameter is the name of a result set to be sorted, or the names of result sets to be merged and the result sorted.

3.2.7.1.2 Sorted-result-set This parameter is the name of the sorted result set. It may be the name of an existing result set (including one of the names included in Input-result-set); if so, then if the sort is processed, the existing result set is deleted, and a new result set by that name is created; its content is the sorted results. If Sorted-result-set is not the name of an existing result set and if the sort is processed, a result set by the specified name is created by the target, whose content is the sorted results; the content of the Input-result-sets is left unchanged. In any case, if the sort is not processed, the final content of Sorted-result-set is indicated by the parameter Result-set-status.

3.2.7.1.3 Sort-sequence The parameter Sort-sequence comprises the elements that are to be used for sorting, together with the direction of the sort (ascending or descending), case sensitivity (if applicable), and target action if an element is missing from a record in the result set to be sorted. Each of the sort elements is a set of attributes, a sort-field-designator, or an element specification, that the target has designated (see note below) for use as a sort key.
Note: The target designates this information either via the Explain facility, or through some mechanism outside of the standard.

3.2.7.1.4 Sort-status The parameter Sort-status, returned by the target, assumes one of the following values:

success	-	The sort was performed successfully.
partial-1	-	The sort was performed but the target encountered records with missing values in one or more sort elements.
failure	-	The sort was not performed. The target supplies one or more diagnostics in the parameter Diagnostics.

3.2.7.1.5 Result-set-status The target supplies this parameter if and only if the value of Sort-status is 'failure'. It refers to the contents of Sorted-result-set, and its value is one of the following:

empty	-	The result set is empty.
interim	-	Partial results available, not necessarily valid.
unchanged	-	The content of the result set is unchanged (applies only if Sorted-result-set is one of the input result sets).
none	-	Result set not created (applies only if Sorted-result-set is not one of the input result sets).

3.2.7.1.6 Diagnostics The target includes this parameter if the value of Sort-status is 'failure'. It includes one or more diagnostic records.

3.2.7.1.7 Other-information This parameter may be used by the origin or target for additional information not specified by the standard.

3.2.7.1.8 Reference-id See 3.4.

3.2.8 Browse Facility

The Browse facility consists of a single service, Scan.

3.2.8.1 Scan Service

The Scan service is used to scan an ordered term-list (subject terms, names, titles, etc.). The ordering of the term-list is target defined. The origin specifies a term-list to scan and a starting term (implicitly, by specifying an attribute/term combination and a database-id), the size of the scanning steps, and the desired number of entries and position of the starting term in the response.

Parameter	Origin Request	Target Response
Database-names	x	
Term-list-and-start-point	x	
Step-size	x (opt)	x (if appl)
Number-of-entries	x	x
Position-in-response	x (opt)	x (opt)
Scan-status		x
Entries		x (opt)
Other-information	x (opt)	x (opt)
Reference-id	x (opt)	x (if appl.)

3.2.8.1.1 Database-names The parameter Database-names identifies a set of databases to which the term-list (specified by Term-list-and-start-point) pertains.

3.2.8.1.2 Term-list-and-start-point The origin supplies an attribute list and term. The attribute list contains attributes indicating which term-list to scan. The term, as qualified by those attributes, indicates where scanning begins; this will be a presumed entry in the term-list. If there is no matching entry, the first entry with higher value is to be the starting point.

As an example, to scan a list of personal names: the attribute list might consist of a single attribute whose type is 'use' and whose value is 'personal name'; the term would specify a personal name; the database-id would identify one or more databases to which the list of personal names pertains.

3.2.8.1.3 Step-size The origin may specify the desired number of entries in the term-list between two adjacent entries in the response. A value of zero means "do not skip any entries." If the target cannot support the requested step size, it sets Scan-status to 'failure' and includes a non-surrogate diagnostic such as "only step size of zero supported" or "requested step size not supported." If the origin omits this parameter, the step size is selected by the target, and the target includes the selected step size in the response.

3.2.8.1.4 Number-of-entries The origin indicates the proposed number of entries to be returned. The target indicates the actual number of entries returned. If the actual number is less than the proposed number, the reason is indicated in Scan-status.

3.2.8.1.5 Position-in-response The origin may optionally indicate the preferred position, within the returned entries, of the specified starting point value. A value of 1 refers to the first of the returned entries. A value of 0 means that the returned entries should begin with the term immediately following the starting point term. A value of Number-of-entries + 1 means that the origin requests terms immediately preceding the starting point term.

The target may indicate the actual position of the chosen starting point within the returned entries. Example: If the values of the request parameters Number-of-entries and Position-in-response are 10 and 3 respectively, then the origin requests two terms immediately preceding the starting point value, followed by the starting point value, followed by the immediately-following seven terms.

Note: If response parameter Position-in-response is less than the value proposed in the request, the origin may conclude that there were fewer terms than expected in the low end of the term-list. However, if Position-in-response is the same value in the response as proposed in the request, but Number-of-entries in the response is less than the value proposed in the request, the origin may not conclude that there were fewer terms than expected at the high end of the term-list, unless Scan-status is Partial-5. The reason that fewer terms than expected are returned is indicated in the Scan-status.

3.2.8.1.6 Scan-status The target indicates the result of the operation. The defined values are:

success	-	The response contains the number of entries (term-list-entries or surrogate diagnostics) requested.
partial-1	-	Not all of the expected entries can be returned because the operation was terminated by access-control.
partial-2	-	Not all of the expected entries will fit in the response message.
partial-3	-	Not all of the expected entries can be returned because the operation was terminated by resource-control, at origin request.
partial-4	-	Not all of the expected entries can be returned because the operation was terminated by resource-control, by target.
partial-5	-	Not all of the expected entries can be returned because the term-list contains fewer entries (from either the low end, high end, or both ends of the term-list) than the number of terms requested.
failure	-	None of the expected entries can be returned. One or more non-surrogate diagnostics is returned.

3.2.8.1.7 Entries The parameter Entries returned by the target:

- Consists of one of the following:
  - N entries, where each entry is a term-list-entry or surrogate diagnostic, where N = Number-of-entries in the request.
  - A number of entries which is less than N, and may be zero (reason specified by Scan-status).
- And may also include:
  - One or more non-surrogate diagnostic records (possibly indicating that the operation cannot be processed, and why it cannot).

Each term-list-entry includes a term (occurring in one of the databases specified in the parameter Database-names), and optionally the following:

- A display term (when the actual term is not considered by the target to be suitable for display).
- A list of suggested attributes for use in subsequent Scan requests (useful for scanning multiple indices, e.g. author and title, at the same time).
- A suggested alternative term.
- Occurrence-information: this might include a count of records in which the term occurs. It may also list counts for specific attributes, possibly further broken down by database. Alternatively, a term-list-entry might list databases in which the term occurs, and for associated attributes, but no counts.
- Other information: additional information concerning the entry.

3.2.8.1.8 Other-information This parameter may be used by the origin or target for additional information, not specified by the standard.

3.2.8.1.9 Reference-id See 3.4.

3.2.9 Extended Services Facility

The Extended Services facility consists of a single service, Extended-services.

3.2.9.1 Extended Services Service

The Extended-Services (ES) service allows an origin to create, modify, or delete a task package at the target. The target maintains task packages in a special database, described in section 3.2.9.2. A task package pertains to an ES task.

An extended service is a task type, related to information retrieval, but not defined as a Z39.50 service. Execution of a task by the target is outside the scope of Z39.50. The extended services defined by this standard are listed in section 3.2.9.1.2. Definitions of those services are included in Appendix 8 EXT.

The origin sends an ES Request to the target requesting execution of a task. The request includes parameters which the target uses to construct the task package. The target checks the request for validity, for consistency with the user's access privileges, and possibly for other target-dependent limitations. The target sends an ES response indicating that the request was accepted or supplying an indication of the reason the request was rejected.

The ES service is a confirmed service, initiated by the origin. The ES operation consists of a request from the origin and a response from the target, possibly with intervening Access-control or Resource-control messages. However, although the request may result in the initiation of a task, the task is not considered part of the Z39.50 ES operation. The target response, which completes the ES operation, does not necessarily signal completion of the task. A task may have a lifetime that exceeds a single Z-association.

Execution of the ES Operation results in the creation of a task package, represented by a database record in the ES database.

For example, when a target creates a task package of type PersistentResultSet, a (persistent) result set is created, represented by the created task package, in the form of a record in the extended services database. When that package is subsequently retrieved by an origin, in either the same or a different Z-association, a copy of that persistent result set is made available to that Z-association, as a Z39.50 result set (i.e. as a transient result set; a result set name, for use during the Z-association, is included within the task package). When an origin deletes the task package, the persistent result set is deleted.

A task package contains parameters, some of which are common to all task packages regardless of package type, and others which are specific to the particular extended service. Among the common parameters (indicated in the table below, listed under "task package parameter" in the right column), some are supplied by the origin as parameters in the ES request, and are used by the target to form the task package; some of those supplied by the origin may be overridden by the target. Others are supplied by the target. The specific parameters are derived from the parameter Task-specific-parameters of the ES request (see Appendix 8 EXT).

Note: The response parameter Task-package below refers to the actual task package, and if it occurs (see 3.2.9.1.13), it includes some or all (depending on the parameter Elements) of the parameters listed under "task package parameter."

Parameter	**Origin request**	**Target response**	**Task Package parameter**
Function	x		
Package-type	x		
Package-name	x (opt)		x (opt)
User-id	x (opt)		x (opt)
Retention-time	x (opt)		x (opt)
Permissions	x (opt)		x (opt)
Description	x (opt)		x (opt)
Target-reference			x (opt)
Creation-date-time			x (opt)
Task-status			x
Package-diagnostics			x (opt)
Task-specific-parameters	x		(see note)
Wait-action	x		
Elements	x (if appl)		
Operation-status		x	
Operation-diagnostics		x (if appl)	
Task-package		x (if appl)	
Other-information	x (opt)	x (opt)	
Reference-id	x (opt)	x (if appl)	

Note: Task-specific-parameters are defined for each extended service. For each task-specific parameter, the definition states whether or not the parameter occurs in the task package.

3.2.9.1.1 Function The origin specifies Create, Delete, or Modify.

If the function is Create, the target is to create a task package, and assign to it the name specified by the parameter Package-name, if supplied.

If the function is Delete or Modify, the target is to delete or modify the task package specified by the parameter Package-name. A target that supports deletion or modification may nonetheless deny the request, for example because the task is already in progress, or the package is in use.

If the function is Delete, the origin requests that if the specified task has not been acted on, it should not be started; if the task is active, the target should either terminate the task or refuse the request.

If the function is Modify, the origin requests that parameter values in the request (as well as those within parameter Task-specific-parameters) replace the corresponding values in the task package. If an optional parameter is omitted, the target does not modify that parameter within the task package (thus to return a parameter to its default value, an origin must explicitly provide the default value).

3.2.9.1.2 Package-type The Package-type identifies the extended service requested. The extended services defined by this standard (see Appendix 8 EXT) are:

- Save a result set for later use
- Save a Query for later use
- Define a periodic search schedule
- Order an item
- Update a database
- Create an export specification
- Invoke a previously created export specification

3.2.9.1.3 Package-name The origin may optionally supply a name for the task package to be created. If so, the triple (Package-type, User-id, Package-name) must be unique (i.e. there must be no other task package of that type, for that user with the same name, otherwise the request is in error), and that triple identifies the task package for subsequent reference. Package-name should be included if the origin intends to reference the task package.

3.2.9.1.4 User-id The User-id identifies the user to be associated with the task package. If not supplied, this parameter may default to the Id of the current user. A target may or may not allow an origin to supply a user id different from its own.

3.2.9.1.5 Retention-time The origin may optionally specify a retention period (e.g., 2 hours, 3 days, 1 week), which may be overridden by the target. When the retention time has passed, the target may delete the retained task package. A retention time of zero means the task package is not to be retained after the task is completed.

3.2.9.1.6 Permissions The origin may indicate who may access the task package. If the origin does not supply this parameter, only the creating user may do so. See 3.2.9.3.

3.2.9.1.7 Description The origin may include a description. It might describe, for example, the result set, for a Persistent Result Set task; or the query, for a Persistent Query task.

3.2.9.1.8 Target-reference The target may supply a unique identifier for the task package.

3.2.9.1.9 Creation-date-time The target supplies the date and time that the task package was created.

3.2.9.1.10 Task-status The target indicates the status of the task. Values are ’pending,’ ’active,’ ’complete,’ and ’aborted’.

3.2.9.1.11 Package-diagnostics The target may include one or more diagnostics in the task package.

3.2.9.1.12 Task-specific-parameters These are additional parameters, defined by the specific extended service.

3.2.9.1.13 Wait-action The origin indicates whether the target should (or may) include the task package in the ES response. This immediate response mechanism may avoid the need for follow-up Search and Present operations, or in general, for making the task package available through the extended services database (see section 3.2.9.2).

This parameter has four possible values:

- *wait:* the target must perform the task before issuing the ES response (unless the operation aborts; see section 3.2.9.4). If the target is not willing to perform the task before issuing the response it must refuse the request by responding with a status of ’failure’ and an appropriate diagnostic. If the target accepts the request, it includes the parameter Task-package in the response.
- *wait-if-possible:* the origin requests that, if possible, the target perform the task before issuing the ES response and include the task package in the response. If not possible, the target should proceed as though the value were ’do not wait’.
- *do-not-wait:* The origin does not request that the target attempt to perform the task before issuing the ES response. However, if the target does perform the task before issuing the response, then the response may include the task package.
- *do-not-send-task-package:* The target may perform the task when it chooses, but is not to include the task package in the response under any circumstance.

3.2.9.1.14 Elements The origin may optionally include this parameter if Wait-action is other than ’do-not-sent-task-package’. It is an element set name for the task package, in the event that it is returned in the response parameter Task-package.

3.2.9.1.15 Operation-status This is the status of the ES operation. It is one of the following:

done	-	The request was accepted, the task is complete and results are included in Task-package.
accepted	-	The request was accepted and the task is queued for processing, or is in process.
failure	-	The request was refused. One or more diagnostics are supplied (in parameter Operation-diagnostics).

3.2.9.1.16 Operation-diagnostics The target may supply additional diagnostic information if Operation-status is ’failure’.

3.2.9.1.17 Task-package If Operation-status is ’done,’ the target includes the task package. The portion of the actual task package included depends on the parameter Elements.

3.2.9.1.18 Other-Information This parameter may be used by the origin or target for additional information, not specified by the standard.

3.2.9.1.19 Reference-id See section 3.4.

3.2.9.2 The Extended Services Database

Targets that support the Extended Services facility provide access to a database with the name IR-Extend-1 (referred to as the "extended services database" or "ES database"). Records in the extended services database are task packages constructed from the Request-parameter-package parameter in ES requests (the target may begin execution of the task at any time after it accepts the request, which may be before the task package has been stored in the database). The target may (but need not) retain a task package until the requested task has completed; it may retain the task package until the origin requests that it be deleted. A target may unilaterally delete a task package from the ES Database at any time.

Note: This means, as a practical matter, the target need not actually create a task package for a given task, in particular, when the task is executed immediately. However, it is recommended that a task package exist when the status of the task is pending, active, or aborted.

When the target receives an ES request it may immediately create a task package, with status ’pending,’ before completely validating the request. The origin may thus search the database anytime after submitting a request (during the same or a subsequent Z-association), for a resulting task package. In particular, if an ES operation is aborted (see 3.2.9.4) the origin may be able to determine that the request for that operation was received.

An ES database may be listed in the target Explain database, with a list of extended services the target supports, allowable export destinations, options that an origin may supply for an export task, etc.

An extended services database will appear to the origin as any other database supported by the target (records may be searched and retrieved by the Z39.50 Search and Retrieval facilities; search processing is defined locally by the target; the target may impose access control or exclude records to which the origin is not authorized access). However, certain search terms are predefined in order to allow a semantic level of interoperability. The attribute set used to search the database is defined and registered in Appendix 3 ATR. The task package structures are defined and registered in Appendix 8 EXT.

The ES database may provide the following special element sets (in addition to "F"):

- Identification: includes the creating user’s identification, the origin-supplied name of the task package, and possible permissions for other users to access the request. Other identifying information such as time of creation may be included.
- UniqueName: the creating user’s identification and the name of the task package.
- Permissions: the contents of the UniqueName element set, and in addition, the granted permissions for the task package. A target might present the full permissions list only to the task package creator, presenting to other users only the permissions applicable to them.
- Status: a short summary of the current status of the request, perhaps including cost and other resource usage.
- Brief: Identification element set plus the most important elements of the Status element set.

3.2.9.3 Owners and Permissions

The creating user of a task package may apply any extended service function to the package, as well as retrieve the full package (via the Retrieval facility) and invoke the package via other extended services. (Invocation occurs, for example, when a Periodic Query task references a saved Query.)

Using the Modify function of the ES request, an origin can change the access permissions of a task package by supplying a new permissions list, which is a sequence of user ids and for each, a sequence of allowed operations, from the following set:

- Delete
- Modify-Contents
- Modify-Permissions
- Present
- Invoke.

As an example of the use of the 'invoke' permission, a target might create a task package, on behalf of a client user, of type PersistentQuery; a persistent query is created, represented by the created task package. The target may subsequently be requested to create a PeriodicQuerySchedule task package, on behalf of a different user, which refers to (i.e. "invokes") that persistent query task package. The target would do so only if that user has 'invoke' privilege for that persistent query. As another example, a target may create an ExportSpecification (package) on behalf of one user, and a different user may subsequently 'invoke' that ExportSpecification by creating an InvokeExportSpecification package, if that user has 'invoke' privilege for the ExportSpecification.

Targets may provide group names for use in permission lists, but a group name would be syntactically the same as a user Id. (The target might report the composition of groups, but the mechanism for doing so is not described by this standard.)

3.2.9.4. Aborted Operations

An origin may receive a response to an ES request only during the Z-association in which it issues the request (as for any other Z39.50 operation). If an ES operation is aborted (explicitly, or because the Z-association is closed or the A-association terminated), the origin will not receive a terminating response. This has no effect on the disposition or processing of the task, regardless of the value of Wait-action that was specified on the request. If an ES operation aborts, Wait-action automatically assumes the value 'do-not-send-task-package'.

If an ES operation is aborted, the origin may search the ES database (possibly in a subsequent Z-association) for information that would otherwise have been returned in the response.

3.2.10 Explain Facility

The Explain facility allows an origin to obtain details of the implementation of a target, including databases available for searching, attribute sets and diagnostic sets used by the target, and schema, record syntax and element specification definitions supported for retrieval. Targets that support the Explain facility:

- provide access (via the Z39.50 Search and Present services) to a database with the name IR-Explain-1 (referred to as the "Explain database");
- support the explain attribute set, exp-1, defined in Appendix 3 ATR (which defines a set of Use attributes and imports bib-1 non-Use attributes); and
- support the Explain syntax, which is defined and registered in Appendix 5 REC.

A record (or result set item representing a record) within the Explain database, is referred to as an "Explain record".

3.2.10.1 Searching the Explain Database

The Explain database appears to the origin as any other database supported by the target. However, certain search terms, corresponding to information categories, are predefined in order to allow a semantic level of interoperability. Terms are searched case-insensitive.

The exp-1 attribute set is used to search the Explain database. Combinations of Use attributes and terms allow searching upon information category; well-defined combinations of Use attributes may be used to allow additional specification by the origin to limit the records to those of immediate interest. Combinations of exp-1 Use attributes to perform a common set of searches are listed in 3.2.10.1.1 and 3.2.10.1.4. Since the Explain database may be searched as any other database using attributes from one or more attribute sets, this list is not exhaustive. However, it is recommended that a target supporting the Explain facility support this list of common searches. As described in 3.2.10.1.2 and 3.2.10.1.3, the HumanStringLanguage, DateAdded, DateChanged, and DateExpires attributes can be used in combination with any of the combinations listed in 3.2.10.1.1 and 3.2.10.1.4.

The exp-1 attribute set consists of a set of Use attributes and imports the non-Use bib-1 attributes. It is recommended that a target supporting the Explain facility support the bib-1 relation attribute 'equal' (see note), position attribute 'any position in field', and structure attribute 'key'.

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

Origins should not in general expect that the explain database is searchable using the bib-1 truncation attribute, completeness attribute nor any of the alternative values of the relation, position and structure attributes defined in bib-1. However, targets are free to provide access to the Explain database using those and other alternative attributes and attribute values.

3.2.10.1.1 Searching for Predefined Information Categories

Records corresponding to a particular explain information category are searched by an operand where the term is the name of that category; for example, all records corresponding to TargetInfo are searched using the term "TargetInfo." For each category one or more *key elements* are defined, and may be provided as search terms (using the appropriate attribute). A search with an operand where the Use attribute = ’ExplainCategory’ and the term is a category, and with additional operands corresponding to each key for that category where the value of the Use attribute is the key, should result in (at most) a single record.

The primary mechanism for search and retrieval of information from the Explain database is for the origin to select the records in a category using the Use attribute ’ExplainCategory’ and to extract desired information from those records to formulate a subsequent search. For example the origin may search records with ExplainCategory = ’DatabaseInfo,’ and retrieve summary information (see 3.2.10.2.2) from those records. Each summary record will include a database name, which serves as a key for a possible subsequent search.

A list and brief description of the Explain information categories (and thus search terms) are given in the table below, as well as the keys for each category. In 3.2.10.3 each category is described in detail.

An origin should adhere to the following rules when searching an Explain database by the predefined information categories.

- To search for information about the target, use ExplainCategory=’TargetInfo’.
- To search for information about a specific database, use ExplainCategory=’DatabaseInfo’ in combination with the DatabaseName attribute to specify the key of the desired databaseInfo record.
- To search for information about a specific schema, use ExplainCategory=’SchemaInfo’ in combination with the SchemaOID attribute to specify the desired schema.
- To search for information about a specific tag set, use ExplainCategory=’TagSetInfo’ in combination with the TagSetOID attribute to specify the desired tag set.
- To search for information about a specific record syntax, use ExplainCategory=’RecordSyntaxInfo’ in combination with the RecordSyntaxOID attribute to specify the desired record syntax.
- To search for information about a specific attribute set, use ExplainCategory=’AttributeSetInfo’ in combination with the AttributeSetOID attribute to specify the desired attribute set.
- To search for information about term lists for a database, use ExplainCategory=’TermList-Info’ in combination with the DatabaseName attribute to specify the desired database.
- To search for information about a specific extended service, use ExplainCategory = ’ExtendedServicesInfo’ in combination with the oid for that extended service.
- To search for the attributes and combination of attributes which may be used in searching a database, use ExplainCategory=’AttributeDetails’ in combination with the DatabaseName attribute to specify the database for which attribute information is desired.
- To search for information about a specific term list, use ExplainCategory = ’TermListDetails’ in combination with the name for the term list.
- To search for the element set names defined for a record syntax for a particular database, use ExplainCategory=’ElementSetDetails’ in combination with the RecordSyntaxOID attribute to specify the desired record syntax and the DatabaseName attribute to specify the desired database.
- To search for the definition of a specific element set name, use ExplainCategory = ’ElementSetDetails’ in combination with the ElementSetName attribute to specify the desired element set name. There may be multiple records located since the explain database contains one record for each element set name for each record syntax for each database.
- To search for a particular element set name defined for a record syntax, for a particular database, use ExplainCategory = ’ElementSetDetails’ in combination with the ElementSetName attribute to specify the desired element set name, the RecordSyntaxOID attribute to specify the desired record syntax and the DatabaseName attribute to specify the desired database.

Category	**An Explain record in this category describes:**	**Key(s)**
TargetInfo	*The target, including search constraints imposed by the target.*	target name
DatabaseInfo	*A database. Information about supported query types, attribute sets, record syntaxes, schemas, diagnostic sets, resource control formats and access control formats. A group of databases offering a common set of characteristics may be described as a single, logical, database. In this case, a list of databases subsumed within this logical database is provided.*	database name
SchemaInfo	*A Schema.*	schema oid
TagSetInfo	*A tag Set.*	tagSet oid
RecordSyntax Info	*A record syntax.*	record syntax oid
AttributeSet Info	*An attribute set, including the attributes supported within the set.*	attribute set oid
TermListInfo	*Term lists supported for a database.*	database name
Extended ServicesInfo	*An extended service.*	extended service oid
Attribute Details	*Attributes that can be used to search a database including the other attributes with which it may be combined.*	database name
TermList Details	*A term list.*	term list name
ElementSet Details	*An element set (for a particular record syntax, for a particular database).*	database name, element set name, record syntax oid
Retrieval RecordDetails	*The elements of a retrieval record (for a particular record syntax, defined by a particular schema).*	databaseName, schema oid, recordSyntax oid
SortDetails	*Sort specification for a database.*	database name
Processing	*Processing instructions for a database, for a particular processing context, name of instructions, and object identifier for the abstract syntax of the externally defined Instructions.*	database name, processing-context, name, oid
VariantSetInfo	*A variant set definition; classes, types, and values, for a specific variant set definition supported by the target. Support by the target of a particular variant set definition does not imply that the definition is supported for any specific database or element.*	variantSet oid
UnitInfo	*Unit definitions supported by the target.*	unit system name
CategoryList	*Explain categories that the target supports.*	(no key)

- To search for the description of the elements of a retrieval record, for a particular record syntax, in a specific schema, for a particular database, use ExplainCategory=’RetrievalRecordDetails’ in combination with the RecordSyntaxOID attribute to specify the desired record syntax, the SchemaOID attribute to specify the desired schema, and the DatabaseName attribute to specify the desired database.

3.2.10.1.2 Searching for Information in a Particular Language

Elements intended to be presented to the user by the origin are said to consist of "human readable text." Each record includes a language element indicating the language of the human readable text within the record. The explain database might contain several records with identical information, in different languages. To search for records in a certain language, the HumanStringLanguage attribute may be used (in conjunction with the three-character language code as the term; see Z39.53-1994).

For example, to search for a list of databases that have descriptive records in English, the query might be of the form:

(Category = ’DatabaseInfo’) AND (HumanStringLanguage = ’eng’).

The HumanStringLanguage attribute is intended primarily for use in Version 2. When version 3 is in force, the use of variants is recommended.

3.2.10.1.3 Searching for Information by Control Dates

To search for new records in an Explain database, use the DateAdded attribute; for updated records use the DateChanged attribute, for records based on their date of expiry use the DateExpires attribute. Any of these three may be used in combination with the searches described above.

3.2.10.1.4 Searching for Information Using Content Values

Some of the Explain records are searchable using attributes which take values from elements within the pertinent Explain records. These Use attributes can be used to select subsets of records of specific information category. For instance, the Availability Use attribute can be used to select those database information records for databases which are currently available. The use of these attributes by an origin should conform to the following rules.

- To locate databases currently available, use the ExplainCategory attribute with term ’DatabaseInfo,’ in combination with the Availability attribute with term ’yes’.
- To locate the databases provided by a specific supplier, use the ExplainCategory attribute with term ’DatabaseInfo,’ in combination with the Supplier attribute with the supplier’s name as term.
- To locate databases provided by a specific producer, use the ExplainCategory attribute with term ’DatabaseInfo’ in combination with the Producer attribute with the producer’s name as term.
- To locate databases that are not proprietary, use the ExplainCategory attribute with term ’DatabaseInfo,’ in combination with the Proprietary attribute with term ’no’.
- To locate databases that have no user fee, use the ExplainCategory attribute with term ’DatabaseInfo,’ in combination with the UserFee attribute with term ’no’.

3.2.10.2 Retrieval of Explain Records

A Present request for Explain records should specify the Explain syntax as the Preferred-record-syntax. Each explain information category has its own record layout, and all are described in the Explain syntax definition (see Appendix 5 REC.1).

Explain records include key elements which serve to uniquely identify each record. Each Explain category is defined in term of key elements, non-key "brief" elements (see 3.2.10.2.2), "non-brief" elements, and possibly other categories. Key elements are always part of the brief elements.

3.2.10.2.1 Retrieval and Human Readable Text

The Explain database might provide alternative variations of human readable information (however, for language variations; see note below). For example, a text element might be retrievable in ASCII, SGML, or Postscript. To request a particular format, use the variant facilities of Version 3.

Note: For language variation, see 3.2.10.1.2. The Explain database logically includes different records for different languages, and therefore selection based on language occurs during the search.

3.2.10.2.2 Retrieving Summary and Descriptive Information

The Explain facility provides for the retrieval of summary, or "brief" information. For example the origin may request summary information about all of the databases supported by a target without retrieving the full databaseInfo records. Within each category's definition, elements are designated as "brief" or "non-brief." Elements designated "brief" are obtained when using the element set name 'B'. Elements designated "non-brief" are obtained (along with brief-elements) when using the element set name 'F'.

The Explain facility also provides for the retrieval of descriptive information, for certain categories, via the element set name 'description' (for details, refer to the ASN.1 definition for the Explain syntax). For example, a Database-info record includes an element which contains a description (in human readable text) of the database; to retrieve only the brief elements and the description element, the element set name 'description' may be used.

Individual categories defined in the Explain syntax may designate other element set names for specific subsets of information within that category.

3.2.10.3 Detailed Descriptions of the Information Categories

This section includes complete descriptions of each information category. In addition to the information enumerated, each record:

- contains information about the record itself, e.g. date of creation and expiration date of the record; and
- includes an element indicating the language of the "human readable text" elements of the record.

These are logical descriptions, which do not reflect the possibility that there might be language variants of a record or syntax variants of a element.

Many of the Explain elements are optional, but are not so indicated in the description below. For specific information, refer to the ASN.1 definition.

3.2.10.3.1 Target-Info Information about the target. There is one such Explain record in the Explain database.

Brief elements:

- A name for the target (only one), in human readable text.
- Recent news of interest to people using this target, in human readable text.
- An icon used to represent this target (in machine presentable form).
- Whether named results sets are supported.
- Whether multiple databases can be searched in one search request.
- The maximum number of concurrent result sets supported.
- The maximum size (in records) of a result set.
- The maximum number of terms allowed in one search request.
- A timeout interval after which the target will trigger an event if no activity has occurred.
- A "welcome" message from the target to be displayed by the origin.

Non-brief elements:

- Contact information for the organization supporting this target.
- A description of the target, in human readable text.
- A set of nicknames or alternate names by which the target is known.
- Restrictions pertaining to this target, in human readable text.
- A payment address (e.g. business office) for the organization supporting this target.
- Hours of operation.
- A list of supported database combinations.
- Internet address and Port number.
- OSI addresses.
- Languages supported for message strings.
- The following elements, where each object listed is supported for one or more databases. (To determine which are supported for a particular database, retrieve the record for that database.)
  - -- Which query-types are supported, and details for each supported type.
  - -- Diagnostic sets supported.
  - -- Attribute sets supported.
  - -- Schemas supported.
  - -- Record syntaxes supported.
  - -- Resource challenges supported.
  - -- Access challenges supported.
  - -- Cost information.
  - -- Variant sets supported.
  - -- element set names supported.
  - -- Unit systems supported.

3.2.10.3.2 Database-Info Detailed description of a database and database-related restrictions and parameters. There is one such Explain record for each database supported.

Brief elements:

- Full database name (only one).
- Whether this is an Explain database (possibly for a different server).
- A list of short (or alternate) names for the database.

- An icon used to represent this database (in machine presentable form).
- Whether there is charge to access this database.
- Whether this database is currently available for access.
- A human-readable name or title for the database (as opposed to the database name, which is typically a short string not meant to be human-readable, and not variable by language.)

Non-brief elements:

- A list of keywords for the database.
- A description of the database, in human readable text.
- Associated databases: those that the target allows (and possibly encourages) to be searched in combination with this database.
- Sub-databases that make up this conceptual single database.
- Any disclaimers concerning this database, in human readable text.
- News about this database, in human readable text.
- A record count for the database (and whether the count is accurate or an estimate).
- A description of the default order in which records are presented, in human readable text.
- An estimate of the average record size (in bytes).
- A maximum record size (in bytes).
- Hours of operation that this database is available.
- Best time to access this database, in human readable text.
- Time of last update of this database.
- Update cycle/interval for this database.
- Coverage dates of this database, in human readable text.
- Whether this database contains proprietary information.
- A description of copyright issues relating to this database, in human readable text.
- A notice concerning copyright which the target expects the origin to display to the user if possible, in human readable text.
- Description and contact information for the database producer, database supplier, and for how to submit material for inclusion in this database, in human readable text.
- Which query-types are supported for this database, and details for each supported type.
- Diagnostic sets supported for this database.
- Attribute sets supported for this database.
- Schemas defined for this database.
- Record syntaxes supported by this database.
- Resource reports supported for this database.
- Text describing access control for this database, in human readable text.
- Costing information related to this database, in both machine readable format, and in human readable text, for connect, present, and search.
- Variant sets supported for this database.
- Element set names supported for this database, with names and descriptions given in human readable text.
- Unit systems supported for this database.

3.2.10.3.3 Schema-Info Descriptive information about a database schema. There is one Explain record for each schema supported by the target.
Note: this is not specific to a database.
Brief elements:

- The object identifier of the schema definition.
- The name of this schema.

Non-brief elements:

- A description of this schema, in human readable text.
- TagSets used by this schema, and for each, a designated tagType.
- The abstract record structure defined by this schema.

3.2.10.3.4 Tag-Set-Info Descriptive information about a given tagSet. There is one such Explain record for each supported tagSet.
Brief elements:

- The object identifier for the tagSet.
- The name of this tagSet.

Non-brief elements:

- A description of this tagSet, in human readable text.
- For each element defined in the tagSet:
  -- The name of the element.
  -- Nicknames for the element.
  -- The tag assigned to the element.
  -- A description of the element.
  -- Its datatype.

3.2.10.3.5 Record-Syntax-Info Descriptive information about a record syntax. There is one Explain record for each abstract record syntax supported by the target.
Note: this is not specific to a database.
Brief elements:

- The object identifier of the abstract record syntax.
- A name by which this syntax is known.

Non-brief elements:

- Transfer syntaxes supported for this abstract syntax (object identifiers).
- A description of this abstract record syntax, in human readable text.
- An ASN.1 module describing the syntax.
- The record structure defined by this syntax.

3.2.10.3.6 Attribute-Set-Info Descriptive information about an attribute set. There is one record for each supported attribute set.

Brief elements:

- The attribute set Id (object identifier) for this attribute set.
- Its name.

Non-brief elements:

- For each attribute type, its name, description, and integer value of the type, and a list of attributes. For each attribute:
  - -- Its name.
  - -- Description.
  - -- Its value.
  - -- Names of equivalent attributes. Equivalences are derived from the attribute set definition (not from the targets behavior).
- Description of the attribute set.

3.2.10.3.7 TermList-Info Descriptive information about term-lists. There is one Explain record for each database.

Brief elements:

- Full database name (one only).
- Summary information about each term-list associated with this database (for each term-list described, there is a TermList-details record):
  - -- Name of the term-list. Must be unique for the database. This is the name to be used to search for the TermList-details record for this term list.
  - -- Its title. For users to see; need not be unique.
  - -- An indication of how expensive it is to search, using the associated attributes. The target indicates one of the following:
    - --- The attribute (combination) associated with this list will do fast searches.

      Note: To obtain the attribute combination, retrieve the associated TermList-details record.
    - --- The attribute (combination) will work as expected. So there is probably an index for the attribute (combination) or some similar mechanism.
    - --- Can use the attribute (combination), but it might not provide satisfactory results. Probably there is no index, or post-processing of records is required.
    - --- cannot search with this attribute (combination) alone.
  - -- Whether the term-list may be scanned.
  - -- A list of names of alternative, broader term-lists.
  - -- A list of names of alternative, narrower term-lists.

(No non-brief elements.)

3.2.10.3.8 Extended-Services-Info Descriptive information about an extended service. There is one Explain record for each extended service supported.

Brief elements:

- The object identifier of the extended service.
- A name by which this extended service is known.
- Boolean flags, indicating:
  - -- Whether it is a private extended service.
  - -- Whether restrictions apply.
  - -- Whether a fee applies.
  - -- Whether the service is available.
  - -- Whether retention is supported.
- What level of wait-action is supported.

Non-brief elements:

- A description, in human readable text.
- Explain elements specific to this extended service (defined within the specific extended service definition).
- An ASN.1 module for the Explain definition.

3.2.10.3.9 Attribute-Details Information for each attribute. There is one Explain record for each supported database.

Brief elements:

- Name of the database to which this attribute information applies.

Non-brief elements:

- For each attribute set supported for the database, the object identifier of the attribute set, and for each attribute within the set:
  - -- The attribute type.
  - -- A default value which applies if the attribute is omitted, and a description of default behavior in human readable form.
  - -- For each value of the attribute:
    - --- The attribute value.

  - --- A description of that value in human readable text.
  - --- Sub-attributes (for Use attributes): a list of alternative values that allow access to the same aspect of the record, but in greater detail.
  - --- Super-attributes (for Use attributes): a list of alternative values that allow access to the same aspect of the record at a coarser level.
  - --- Whether the value is only "partially supported": i.e. the value is accepted but may not provide expected results.
- A list of all attributes combinations supported for the database.

3.2.10.3.10 Term-list-Details Descriptive information for a term-list. There is one record for each term-list listed by TermList-info records.
Brief elements:
- Name of the term-list.

Non-brief elements:
- A description.
- Attribute combination corresponding to this list. If list may be scanned, this is the attribute combination to be used by scan.
- Maximum step-size supported.
- Collating sequence (e.g. ASCII, EBCDIC) in human-readable text.
- Order (ascending or descending).
- Estimated number of terms.
- A list of sample terms (not guaranteed to be valid; optimally would represent a uniformly distributed sampling of the list).

3.2.10.3.11 Element-Set-Details Descriptive information about an element set. There is one Explain record for each element set for each record syntax for each database.
Brief elements:
- The database to which this record pertains.
- The element set name for the element set described by this record.
- The record syntax to which this record pertains.
- The schema for which this element set is defined.

Non-brief elements:
- A description, in human readable text, of the element set.
- For each element in the element set, the information provided for each element by the Retrieval-Record-Details category.

3.2.10.3.12 Retrieval-Record-Details Descriptive information about the elements of a retrieval record. Note that the elements are relative to a database schema. There is one such Explain record for each database for each schema for each record syntax.
Brief elements:
- The database, schema, and record syntax to which this Explain record pertains.

Non-brief elements (for each element described by the syntax):
- The name of the element.
- The tag of the element, if any.
- A list of schema elements that comprise this element within the record syntax.
- The maximum size of the element.
- The minimum size of the element.
- The average size of the element.
- The size of the element, if fixed length.
- Whether or not the element is repeatable.
- Whether or not the element is required.
- A description of the element, in human readable text.
- A description of its contents, in human readable text.
- Charging/billing issues related to this element, in human readable text.
- Restrictions (e.g. copyright, proprietary) pertaining to use and access to this element, in human readable text.
- Alternate names for this element.
- Generic names for this element text (e.g. a "geographicSubject" element might also be under the generic name "subject").
- Attribute combinations corresponding to this element.

3.2.10.3.13 Sort-Details Description of the sorting capabilities supported by the target. There is one record for each database.
Brief elements:
- Database to which this sort description pertains.

Non-brief elements:
- For each sort key:
  - -- A description.
  - -- If the key is a record element, a specification of the element.
  - -- If the key is an attribute combination, a specification of that combination.
  - -- The type of key: character, numeric, structured.
  - -- Whether the key is case-sensitive.

3.2.10.3.14 Processing-Info Instructions, representing how the target believes the data should be processed by the origin for presentation to the user. Instructions are defined externally. For a given database and processing context (access, search, retrieval, record-presentation, and record-handling) for which the target offers processing information, there may be more than one set of instructions; these are distinguished by name. Each set of instructions may be available in more than one abstract syntax; these are distinguished by object identifier. Thus an Explain record of this type is distinguished by database, processing context, name, and object identifier.
Brief elements:

- Full name of the database to which this record pertains.
- The context for which this processing information is pertinent.
- A name for this processing information.
- An object identifier, for the abstract syntax of the externally defined instructions.

Non-brief elements:

- A description of the instructions, in human readable form.
- The machine processable instructions, externally defined (whose abstract syntax is identified by the object identifier referenced above).

3.2.10.3.15 Variant-set-info Descriptive information about a variant set definition supported by the target; classes, types, and values supported for a particular variant set. Support of a particular variant set definition does not imply that the definition is supported for any specific database or element.
Brief elements:

- The object identifier of the variant set definition.
- Its name.

Non-brief elements:

- A list of supported classes, including name and description; and for each, a list of supported types, including name and description; and for each, a list of supported values.

3.2.10.3.16 Unit-info Descriptive information about a unit system definition supported by the target.
Brief elements:

- The name of the unit system.

Non-brief elements:

- A description.
- A list of unit types, including name and description, and for each, a list of units, including name and description.

3.2.10.3.17 Category-list A list of the Explain categories supported by the target. There is one such record for the Explain database. It consists of the information below, for each supported category.
Brief elements:

- The search term used in conjunction with Use attribute of ExplainCategory to search for records of this category.

Note: the following need occur only if the target is supporting a category not defined in this standard.

- The original search term. (This is for information categories where the target is supporting a revision of the original definition of a category.)
- A description.
- An ASN.1 definition of the record for this category.

3.2.11 Termination Facility

The Termination Facility consists of the single service, Close.

3.2.11.1 Close Service

The Close service allows either an origin or target to abruptly terminate all active operations and to initiate termination of the Z-association.

The Close service may be used only when version 3 is in force. If so, following initialization, at any time until a Close request is either issued or received, either the origin or target:

- may issue a Close request, consider all active operations to be abruptly terminated, await a Close response (discarding any intervening messages), and consider the Z-association closed; and
- should be prepared to receive a Close request, consider all active operations to be abruptly terminated, issue a Close response, and consider the Z-association closed.

Parameter	**Request**	**Response**	**Note**
Close-reason	x	x	
Diagnostic-information	x (opt)	x (opt)	
Resource-report-format	x (opt)		Origin only
Resource-report	x (opt)	x (opt)	Target only
Other-information	x (opt)	x (opt)	
Reference-id	x (if appl)	x (if appl)	

3.2.11.1.1 Close-reason This parameter indicates the reason why the origin or target is closing the Z-association. Its values are:

- finished
- shutdown
- system problem
- cost limits
- resources
- security violation
- protocol error
- lack of activity
- unspecified
- response to Close request

Note: Both the Close request and Close response map to the same protocol message (Close APDU). If both systems issue a Close request at the same time, each will receive the peer message as a Close response (even though the message was not sent as such). This potential ambiguity will not effect the correct operation of the protocol. However, for the case where the message is indeed sent as a Close response, the last of the above listed statuses, "response to Close request" is provided and may optionally be used.

3.2.11.1.2 Diagnostic-information The target may include an optional text message, providing additional diagnostic information.

3.2.11.1.3 Resource-report-format and Resource-report When the origin issues a Close request: the origin may include the parameter resource-report-format to request that the target include a resource report (see 3.2.6.1.1) in the response. The target's decision to include a resource report in the response (and the format) is unilateral: it may include or omit a report regardless of whether the origin included the parameter resource-report-format.

When the target issues a Close request: the target may unilaterally include a resource report.

3.2.11.1.4 Other-information This parameter may be used by the origin or target for additional information, not specified by the standard.

3.2.11.1.5 Reference-id The parameter Reference-id may be included or omitted on a Close request or response from the origin.

The target should omit Reference-id on a Close request. On a Close response, if the target is responding to a Close request that included Reference-id, the target may either include Reference-id using the identical value, or it may omit the parameter. If the target is responding to a Close request that did not include a Reference-id, the target should omit the parameter.

3.3 Message/Record Size and Segmentation

A "segment" is a message that is sent (or is in preparation for transmission) by the target as part of an aggregate Present response, i.e., a Segment request or Present response.

Throughout 3.3, "record" is used as follows:

- Unless otherwise qualified, it means "response record," i.e., retrieval record or surrogate diagnostic.
- Except within 3.3.3, it means "surrogate diagnostic record" if the record size exceeds preferred-message-size.
- "Record N" means "the response record corresponding to the database record identified by result set entry N."
- A record is considered to be a string of bytes (for the purpose of describing segmentation procedures).
- "Record size" refers to the size of a record, in bytes.

Except within 3.3.3, a set of records is said to "fit in a segment" if the sum of their sizes, not including protocol control information, does not exceed Preferred-message-size. For the Present operation, the target might be unable to fit the requested records in a single segment, because of record or message size limitations. In that case, the target may perform segmentation of the Present response (if segmentation is in effect) by sending multiple segments (Segment requests followed by Present response).

Two levels of segmentation, level 1 and level 2, are subject to negotiation. If neither level is in effect, the target response to a Present request consists of a simple Present response (a single segment) which contains an integral number of records. If level 1 segmentation is in effect, the target response to a Present request may consist of multiple segments (Segment requests followed by a Present response), and each segment must contain an integral number of records, i.e., records may not span segments. If level 2 segmentation is in effect, the target response to a Present request may consist of multiple segments, and records may span segments.

3.3.1 Procedures When No Segmentation is in Effect

The procedures in this section (3.3.1) apply when no segmentation is in effect. (They apply not only to a Present operation when no segmentation is in effect, but they also apply in general to a Search operation, whether or not segmentation is in effect; a Search response is not subject to segmentation.)

The target responds to a Present request with a simple Present response (or to a Search request with a Search response), which contains an integral number of records. If the target is not able to return all of the records requested, because of message size limitations, the target should fit as many records as possible.

Assume that the target is attempting to return records M through N. If records M through N fit in the response, then the target returns those records. Otherwise, the target returns records M through P, where P is chosen such that records M through P fit in the response, but records M through P+1 do not.

Illustration

Assume that the target is attempting to return records 1 through 10; records 1 through 6 fit in the response, but retrieval records 1 through 7 will not fit.

The size of retrieval record 7, itself:

(a) does not exceed Preferred-message-size, or
(b) exceeds Preferred-message-size, but does not exceed Exceptional-record-size, or
(c) exceeds Exceptional-record-size.

In case (a), the target returns records 1 through 6. In case (b), except as noted below (see "Exception"), the target substitutes a diagnostic record for retrieval record 7, indicating that the record exceeds Preferred-message-size. In case (c) the target substitutes a diagnostic record for retrieval record 7, indicating that the record exceeds Exceptional-record-size. (If Exceptional-record-size equals Preferred-message-size then there is no distinction between the meaning of the two diagnostics.)

In case (b) or (c):

- If the diagnostic record will not fit along with records 1 through 6, the target returns records 1 through 6. (Preferred-message-size must always be large enough to contain any diagnostic record; thus a subsequent present request beginning with record 7 will retrieve the diagnostic.)
- Otherwise, the target inserts the diagnostic record and proceeds to attempt to fit records 8 through 10.

Exception

If a Present request specifies a single record (i.e. Number-of-records-requested equals 1) then if the size of that retrieval record exceeds Preferred-message-size, but does not exceed Exceptional-record-size, the target will return that single retrieval record. Note that this exception applies only to a Present operation and not to a Search operation.

Thus in case (b), the origin may subsequently retrieve retrieval record 7, by issuing a Present request in which that record is the only record requested.

Note that the purpose of this distinction between Preferred-message-size and Exceptional-record-size is to allow the transfer of normal length records to proceed in a routine fashion with convenient buffer sizes, while also providing for the transfer of an occasional exceptionally large retrieval record without requiring the origin to continually allocate and hold local buffer space for worst-case records. Note also that this intended purpose is defeated if the origin routinely requests a single record.

3.3.2 Level 1 Segmentation

When level 1 segmentation is in effect, the target may segment the aggregate Present response into multiple segments (zero or more Segment requests followed by a Present response), each consisting of integral records (i.e. records may not span segments). The procedures described in this section (3.3.2) apply if level 1 segmentation is in effect.

Beginning with the first record requested and continuing with adjacent higher number records, the target forms segments to contain the requested records. Each segment is sent as a Segment request, except the last, which is sent as a Present response.

The number of segments must not exceed the value of the (optional) Present request parameter Max-segment-count, if supplied.

If Max-segment-count is supplied, and its value is 1, then the procedures of 3.3.1 apply. Also, the same exception as cited in 3.3.1 applies if a Present request has requested a single record.

Assume that the origin requests result set records M through N.

Case A: M<N (i.e. more than one record requested).

1. Set P=M.
2. If records P through N fit in a segment:
   - Fit records P through N in the segment.
   - Go to step 3.

   Otherwise,
   - Fit records P through Q, where Q (which is less than N) is such that records P through Q

fit in a segment, but records P through Q+1 do not.
- If Max-segment-count is reached, go to step 3.
- Send the segment as a Segment request.
- Set P=Q+1.
- Repeat step 2.

3. Send the segment as a Present response.

Case B: M=N (i.e. a single record requested).
The target sends a simple Present response (a single segment). The size of the segment may exceed Preferred-message-size. The segment contains the single requested retrieval record, or a surrogate diagnostic record if the size of the record exceeds Exceptional-record-size.

Illustration

Assume the origin has requested records 1 through 10.

1. If all ten records fit in a segment, the aggregate Present response consists of a Present response including the requested records. Present-status is 'success' (all expected response records available).
2. Suppose records 1 through 4 fit in a segment, but records 1 through 5 do not; records 5 through 9 fit in a segment but records 5 through 10 do not. (Assume the Present request has specified a value of 3 or greater for the parameter Max-segment-count.) Then the aggregate Present response consists of:
   - a segment request including records 1 through 4,
   - a segment request including records 5 through 9, and
   - a Present response including record 10.

   Present-status is 'success' (all expected response records available).

   Note that the target is expected to pack as many records into a segment as will fit; thus for example, the first segment would not consist of records 1 through 3, because records 1 through 4 will fit.
3. Assume the conditions in (2) are true, except that the Present request has specified a value of 2 for the parameter Max-segment-count. Then the aggregate Present response consists of:
   - a Segment request including records 1 through 4, and
   - a Present response including records 5 through 9.

Present-status is 'partial-2' (not all expected response records available, because they will not all fit within the preferred message size).

3.3.3 Level 2 Segmentation

When level 2 segmentation is in effect, the target may segment the aggregate Present response into multiple segments (as is the case for level 1 segmentation) and in addition, records may span segments. The procedures described in this section (3.3.3) apply if level 2 segmentation is in effect.

If a retrieval record will not fit in a segment (along with records already packed into the segment) it may be segmented into multiple contiguous fragments (see 3.3.3.1) to be packed into consecutive segments according to the procedures detailed in 3.3.3.2 and 3.3.3.3.

3.3.3.1 Fragments

A fragment is a proper substring of a record (as noted above, within section 3.3.3 a record is treated as a string of bytes). A particular instance of segmentation of a record results in a sequence of two or more fragments whose concatenation (not including protocol control information) is identical to the record. However, there may be different instances of segmentation of a particular record, and the origin cannot necessarily predict how a record will be segmented into fragments by the target in a particular instance.

For the purpose of procedure description (3.3.3.3) a starting fragment is defined to be a fragment that starts at the beginning of a record. An intermediate fragment is a fragment that neither starts at the beginning nor ends at the end of a record. A final fragment is a fragment that ends at the end of a record. An integral record (not segmented) is not a fragment.

The sum of the sizes of the records and record fragments in a segment, not including protocol control information, must not exceed Max-segment-size (see 3.3.3.2).

3.3.3.2 Segment Size, Record Size, and Segment Count

If level 2 segmentation is in effect, the Present request may optionally include these three parameters:
Max-segment-size -- The largest allowable segment. If included, overrides Preferred-message-size (for this Present operation only). If not included, Max-segment-size assumes the value Preferred-message-size.
Max-record-size -- The largest allowable retrieval record within the aggregate Present response. If included, it must equal or exceed Max-segment-size. (If level 2 segmentation is in effect, the parameter Exceptional-record-size that was negotiated during initialization does not apply, whether or not Max-record-size is included, unless the value of Max-segment-count is 1.)

Max-segment-count -- The maximum number of segments the target may include in the aggregate Present response. If its value is 1, no segmentation is applied for the operation, the procedures of section 3.3.1 apply, and Max-record-size should not be included.

If the latter two parameters are both included, Max-record-size must not exceed the product Max-segment-size times Max-segment-count.

If Max-record-size but not Max-segment-count is included, the origin should be prepared to receive as many segments as necessary to retrieve the requested records.

If Max-segment-count is included (and its value is greater than 1), but Max-record-size is not, the product Max-segment-size times Max-segment-count is the maximum record size for the operation.

If the latter two parameters are both omitted the origin should be prepared to receive arbitrarily large records and an arbitrary number of segments.

3.3.3.3 Segmentation Procedures

The following procedures apply for level 2 segmentation. The target fits as many integral records as possible into the first segment. If all of the requested records will fit, the segment is sent as a simple Present response. Otherwise, in the space remaining within that segment the target fits a starting fragment of the following record (if possible), and the segment is sent as a Segment request. The target then fits the remainder of that record into the next segment (if possible; and if not possible, sends Segment requests as necessary with intermediate fragments, and fits the final fragment, if any, into the beginning of the next segment) and fills as many integral records as possible within the space remaining within that segment. If the last of the requested records is placed in the segment (or Max-segment-count is reached) the segment is sent as a Present response. Otherwise the target continues to fill segments in this manner until the last of the requested records is placed in a segment or Max-segment-count is reached, and sends each segment as a Segment request except the last, which it sends as a Present response. These procedures are re-iterated more formally as follows:

Assume that the origin requests records M through N. (Note that "Record" means "surrogate diagnostic record" if the size of the record exceeds Max-record-size, or if the target is unable to segment the record so that each fragment fits within a segment.)

1. Set R=M (begin preparation of first segment).
2. If record R fits in the current segment:
   - Fit integral records R through P, where P is the largest number (not exceeding N) so that records R through P fit.
   - If P equals N, or if Max-segment-count is reached, go to step 8.
   - R=P+1
3. *Note*: having reached this step, record R will not fit in the current segment.
   If the Present request has included Max-segment-count and the target is unable to determine whether record R will fit in the remainder of the aggregate response:
   - Insert a surrogate diagnostic record, which in effect suggests that the origin might again attempt to retrieve the record, but without specifying a Max-segment-count.
   - Go to step 7.
4. If record R will not fit in the remainder of the aggregate response, go to step 8.
5. If record R will fit in the remainder of the aggregate response, but no starting fragment will fit in the current segment:
   Note: this condition precludes the possibility that the segment is empty; see note preceding step 1.
   - Transmit a Segment request (begin preparation of the next segment).
   - go to step 2.
6. Note: having reached this step, Record R will fit in the remaining segments; it will not fit within the current segment, but a starting fragment will fit in the current segment.
   - Fit the largest possible starting fragment of record R and transmit a Segment request.
   - Fill as many complete segments as necessary (which may be zero) with intermediate fragments of record R and send Segment requests.
   - Begin preparation of the next segment, first inserting the final fragment of record R.
7. - Set R = R+1.
   - If R is less than or equal to N, go to step 2.
8. Send a Present response.

Illustration:

Assume the origin has requested records 1 through 12. All records are 500 bytes, except record 5, which is 10,000 bytes. Max-segment-size is 3200.

1. Suppose record 5 consists of 10 elements, each 1000 bytes. The target is able to segment record 5, but only at element boundaries; the target will not let the elements span fragments.

Note: this means that the target may segment the record so that a fragment consists of bytes M*1000+1 through (M+N)*1000, M= 0,1, ...9; N = 1, 2, ..., 10-M; e.g. bytes 1-1000, 1-2000, 1-3000, 1000-2000, 1000-3000, 1000-4000, etc.
Suppose further that the target cannot segment any other records. The aggregate Present response is as follows:

segment 1:	Segment request consisting of records 1 through 4, and the first 1000 bytes of record 5 as a starting fragment.

Note: the size of the segment is 3000 bytes which is less than the Max-segment-size of 3200; but the target cannot fit another fragment in the segment because that would cause the segment size to exceed the 3200-byte maximum (the minimum fragment size is 1000 bytes).

segment 2:	Segment request consisting of bytes 1001 through 4000 of record 5 as an intermediate fragment.
segment 3:	Segment request consisting of bytes 4001 through 7000 of record 5 as an intermediate fragment.
segment 4:	Segment request consisting of bytes 7001 through 10,000 of record 5 as a final fragment.
segment 5:	Segment request consisting of records 6 through 11.
segment 6:	Present response consisting of record 12.

2. Suppose further that the target can segment the smaller records into 100 byte fragments (or multiples).
Segments 1 through 3 are as in case 1.

segment 4:	Segment request consisting of bytes 7001 through 10,000 of record 5 as a final fragment, and bytes 1 through 200 of record 6 as a starting fragment.
segment 5:	Segment request consisting of bytes 201 through 500 of record 6 as a final fragment, records 7 through 11, and the first 400 bytes of record 12 as a starting fragment.
segment 6:	Present response consisting of bytes 401 through 500 of record 12 as a final fragment.

3. Suppose the target can segment any of the records at arbitrary byte boundaries.

segment 1:	Segment request consisting of records 1 through 4 and the first 1200 bytes of record 5 as a starting fragment.
segment 2:	Segment request consisting of bytes 1201 through 4200 of record 5 as an intermediate fragment.
segment 3:	Segment request consisting of bytes 4201 through 7400 of record 5 as an intermediate fragment.
segment 4:	Segment request consisting of bytes 7401 through 10,000 of record 5 as a final fragment, record 6, and the first 100 bytes of record 7 as a starting fragment.
segment 5:	Present response consisting of bytes 101 through 500 of record 7 as a final fragment, and records 8 through 12.

3.4 Operations and Reference-id

A request from the origin of a particular operation type initiates an operation, which is terminated by the respective response from the target. The following operation types are defined: Init, Search, Present, Delete, Resource-report, Sort, Scan, and Extended-services. (Thus each origin request type corresponds to an operation type with the exception of the following request types: Trigger-resource-control and Close.) An operation consists of the initiating request and the terminating response, along with any intervening Access-control and Resource-control requests and responses, Trigger-resource-control requests, and Segment requests. An operation is assigned a Reference-id by the origin, the origin includes the Reference-id within the initiating request, and it must be included within each message of the operation. If 'serial operations' is in effect, the Reference-id parameter may be omitted in the initiating request; in that case the reference-id is considered null for that operation, and all other messages of that operation must also omit the Reference-id parameter.

Any message sent from origin to target or vice versa (i.e. any request or response defined by this service definition) is part of an operation (identified by its Reference-id), with the following exceptions:

- A Close request or response is not part of any operation.
  Note: A Close request or response may include a reference-id, according to the procedures specified in 3.2.11.1.5.

- If ’concurrent operations’ is in effect any Resource-control or Access-control request or response which does not include a Reference-id is not part of an operation.

This standard does not assume any relationship between a given operation and any subsequent operation even if the latter operation uses the same reference-id. This standard does not specify the contents of the Reference-id parameter, nor its meaning, except to the extent that it is used to refer to an operation. Reference-ids are always assigned by the origin and have meaning only within the origin system. Since no semantics are attributed to the Reference-id, it has no implied data type and can only be described as transparent binary data. (Its ASN.1 type is therefore OCTET STRING.)

3.5 Concurrent Operations

If ’concurrent operations’ is in effect, the Reference-id parameter is mandatory in an initiating request (however, see note), and the origin may initiate multiple concurrent operations, each identified by a different reference-id.

Note: The Reference-id parameter is always optional in an Init request; ’concurrent operations’ does not take effect until negotiation is complete, and is thus not in effect during an Init operation.

Once an operation is initiated, until that operation is terminated, another operation may not be initiated with the same reference-id. This standard does not specify the order in which concurrent operations are processed at the target; the target may process concurrent operations in any manner it chooses.

Example:

the origin may issue a Search request using Reference-id "100," and then issue a second Search request using Reference-id "101" before receiving the Search response from the first Search request. There would then be two concurrent operations. Receipt by the origin of the response corresponding to the second Search request (identified by Reference-id "101") would terminate the second operation, and that might occur before termination of the first operation (identified by Reference-id "100"). The origin might then issue a Present request (against the result set created by the second operation), initiating another operation. In that case, the origin must supply a Reference-id other than "100" (because there is an active operation with that Reference-id). The new Reference-id could (but need not) be "101"; if it is, the target may not assume any implied relationship between this new operation and the previous operation which used Reference-id "101."

No operation may be initiated while an Init operation is in progress. No operation may be initiated within a Z-association after a Close request has been sent or received.

All result sets are, in principle, available to any operation. It is possible that two or more concurrent operations will attempt to reference the same result set. This standard does not specify what happens in that circumstance. The origin should not initiate concurrent Search operations with the same value of Result-set-id.

Other than the restriction cited above (that when the origin uses a Reference-id to initiate an operation, until that operation is terminated it may not use that Reference-id to initiate another operation) there are no restrictions on the re-use or management of Reference-ids by the origin. The origin might re-cycle Reference-ids randomly among users, or it may manage local threads by assigning different Reference-ids to end-users. The target is not required to know how the origin manages Reference-ids, or in particular, that the origin is using Reference-ids to distinguish different users. There is no requirement for the target to have any knowledge of multiple end users at the origin, the target interacts only with the (single) origin.

3.6 Composition Specification

For each database supported the target defines one or more schemas (see 3.1.5), and designates one as the default schema. For each schema, the target designates one or more element specification identifiers.

An element specification identifier is the object identifier of an element specification format (a structure used to express an element specification) or an element set name. The latter is a primitive name. An element specification is an instance of an element specification format, or an element set name.

For the default schema, at least one of the element specification identifiers must be an element set name, and the target designates one as the default element set name for the database.

Note: the target designates this information either via the Explain facility, or through some mechanism outside of the standard.

For each record to be returned in a Search or aggregate Present response, the target applies an abstract record structure (defined by a schema for the database to which that record belongs) to form an abstract database record, to which the target applies an element specification to form another instance of the abstract database record (the latter might be a null transformation), to which the target applies a record syntax, to form a retrieval record.

If the origin includes the parameter Comp-spec (in a Present request) the procedures of 3.6.1 apply. For a Search operation, or a Present operation when the parameter Comp-spec is omitted, the default schema is assumed for each record, and the procedures of 3.6.2 apply.

3.6.1 Comp-spec Specified

The Present request parameter Comp-spec includes a set of one or more pairs of a database name and associated composition specification. Each composition specification may include a schema identifier (or if not, the default schema for the database is assumed) and an element specification. For each record to be returned in the aggregate Present response:

- If the database to which the record belongs is specified (as a component of one of the pairs) then the target forms an abstract database record by applying the corresponding composition specification (i.e. by first applying the abstract record structure, defined by the schema, to the database record to form an abstract database record, and then applying the element specification; where the schema and element specification are from the composition specification), if it is able to do so.
- Otherwise, the target forms an abstract database record by applying the abstract record structure defined by the default schema, and default element set name, for the database to which the record belongs.

The parameter Comp-spec may alternatively consist of a single composition specification with no database specified. In that case, for each record to be returned, if the target is able to form an abstract database record according to that composition specification, it does so. If not, an abstract database record is composed according to the default schema and default element set name for the database to which the record belongs.

The target applies a record syntax (which may be included in the composition specification or within the parameter Preferred-record-syntax) to the resulting abstract database record, to form a retrieval record.

3.6.2 Comp-spec Omitted

When requesting the retrieval of a set of records from a result set, if the parameter Comp-spec is omitted, the procedures of this section apply.

Notes:

1. This is always the case on a Search request, because the parameter Comp-spec is not included in the definition of the Search request.
2. This is always the case when version 2 is in force, because the parameter Comp-spec is not defined in version 2.

The Search request parameters Small-set-element-set-names and Medium-set-element-set-names, and the Present request parameter Element-set-names, take the form of a set of one or more pairs of a database name and associated element set name. For each record to be returned in the Search or aggregate Present response, the target first applies the abstract record structure defined by the default schema for the database to which the record belongs, to form an abstract database record, and then applies an element set name, as follows:

- If the database to which the record belongs is specified (as a component of one of the pairs), and if the corresponding element set name is valid for the default schema for the database, then the target applies that element set name.
- If not, the target applies the default element set name for the database.

Each of these parameters may alternatively consist of a single element set name with no database specified. In that case, for each record to be returned, if the element set name is valid for the default schema for the database to which the record belongs, the target applies that element set name; if not, the target applies the default element set name for the database.

A target must always recognize the character string "F" as an element set name to mean "full"; when it is applied to an abstract database record, it results in the same abstract database record (i.e. a null transformation).

A target must always recognize the character string "B" as an element set name to mean "brief" record. This standard does not define the meaning of "brief." Unless the origin knows the target’s definition of "brief" for a given schema, it should not assume that any particular elements are included.

The origin may specify a "preferred-record-syntax," which the target applies (to the abstract database record formed by the application of the element set name) to form a retrieval record. If the origin does not specify a preferred-record-syntax the target may select one (see 3.2.2.1.5).

3.6.3 Record Syntax

For each record to be returned in a Search or aggregate Present response, the element set name, or the schema and element specification from the composition specification, results in an abstract database record, as described above. To that abstract database record, the target applies a record syntax, indicated as described above. The term "record syntax" has the following meaning:

- When specified by the origin (either as the value of Preferred-record-syntax or within a composition specification), it takes the form of an OID and refers to an abstract syntax (paired, or to be paired by the target, with a transfer syntax) that the origin requests the target use for retrieval records.
- When specified by the target, it takes the form of an OID or p-context accompanying a retrieval record in a Search or Present response, and it refers to an abstract syntax paired with a transfer syntax.

3.7 Type-1 and type-101 Queries

This section specifies procedures when Query-type is 1 (or 101; see Note 2 below). Type-1 is the "Reverse Polish Notation" (RPN) query. It has the following structure:

```
RPN-Query ::= Argument
                     | Argument + Argument + Operator
    Argument  ::=     Operand | RPN-Query
    operand   ::=     AttributeList + Term
                          | ResultSetId | Restriction
    Restriction ::=   ResultSetId + AttributeList
    operator  ::=     AND | OR | AND-NOT | Prox
```

The notation above is used as follows:

::=	means "is defined as"
|	means "or"
+	means "followed by", and + has precedence over | (i.e., + is evaluated before |).

Notes:

1. For type-1, the Prox operator and the Restriction operand are defined for version 3 only. When version 2 is in effect, it is a protocol error to include either the Prox operator or Restriction operand in a type-1 query.
2. The type-101 query is defined as identical to the type-1 query, with the exception that the Prox operator and Restriction operand are defined not only for version 3, but for version 2 as well. Thus the definition of the type-101 query is independent of version.

A Z39.50-conforming target must support the type-1 query, but support of the type-1 query does not imply support of any of the defined operators or operands.

The target designates what query types it supports, and which operators and operands.

Note: The target designates this information either through the Explain facility or through some mechanism outside of the standard.

If the target claims support for the Prox operator, the target should also designate whether it supports the extended result set model for proximity (the extended result set model for searching as described in 3.1.6 and its specialization for proximity as described in 3.7.2.2). If the target claims support for the Restriction operand, then it must also support the extended result set model for restriction (the extended result set model for searching and its specialization for restriction as described in 3.7.3).

Note: Only in certain circumstances (detailed below) does support of the Prox operator require support of the extended result set model for proximity. However, support of the Restriction operand always requires support of the extended result set model for Restriction.

3.7.1 Representation and Evaluation of the Type-1 and Type-101 Queries

At the origin, the query is represented by a tree. Each subtree represents an operand, either a simple operand or a complex operand. Each leaf node represents a simple operand: Result-set-id, AttributeList+ Term, or Restriction. Each non-leaf node represents a complex operand: a subtree whose root is an operator, and which contains two subtrees, a left operand and a right operand.

The origin traverses the tree according to a left post-order traversal, to produce a sequence of (simple) operands and operators, which is transmitted to the target.

At the target, evaluation of the sequence of operands and operators is illustrated by the use of a stack. Whenever an operand is encountered, it is put on the stack. Whenever an operator is encountered, the last two objects that have been put on the stack are pulled off and the operator is applied as follows.

Each operand represents a set of database records. Each is one of the following:

(a) AttributeList+term -- in which case it represents the set of database records obtained by evaluating the specified attribute-set and term against the collection of databases specified in the Search request.

(b) ResultSetId -- in which case it represents the set of database records represented by the transient result set identified by ResultSetId.

(c) Restriction operand (ResultSetId+AttributeList): in which case it represents the set of database records represented by the result set identified by ResultSetId, restricted by the specified attribute set (see 3.7.3).

Note: if the Restriction operand occurs the target must support the extended result set model for restriction; otherwise the query is in error.

(d) An intermediate result set (resulting from a previous evaluation placed on the stack) -- in which case it represents the records identified by that result set.

Let S1 and S2 be the sets represented by the left and right operand respectively. Let S be defined as follows:

- If the operator is AND, S is the intersection of S1 and S2.
- If the operator is OR, S is the union of S1 and S2.
- If the operator is AND-NOT, S is the set of elements in S1 which are not in S2.
- If the operator is Prox:
  - If both operands are of form (a) S is the subset of records in the set (S1 AND S2) for which A ProxTest B is true (see 3.7.2.1) where A and B are the two operands.
  - Otherwise:
    - -- The target must support the extended result set model for proximity; or else the query is in error.
    - -- let R1 and R2 be result sets representing the sets S1 and S2

      (i.e., each is either:

      --- the result set specified by the corresponding operand, if it was of form (b), or

      --- the hypothetical result set representing the set of records represented by that operand, otherwise.

      In either case, both R1 and R2 are assumed to conform to the extended result set model for proximity.)

      Each entry in R1 and R2 contain positional information, in the form of position vectors. For each record represented by both R1 and R2, consider every ordered pair consisting of a position vector associated with the record as represented in R1 and a position vector associated with the record as represented in R2. For each pair that qualifies according to the ProxTest:

      --- the record is qualified into the set S; and

      --- a position vector is created for that record as represented in the resultant set, composed from that ordered pair.

An intermediate result set is created, which represents the records in the set S, and is put on the stack. When evaluation of the query is complete (i.e. all query-terms have been processed) there will be one object remaining on the stack (otherwise the query is in error), representing a set of database records, which is the result of the query.

3.7.2 Proximity

3.7.2.1 The Proximity Test

The proximity test, ProxTest, includes a Distance, Relation, Unit, and two boolean flags: Ordered and Exclusion.

- Distance: Difference between the ordinal positional values of the two operands. (E.g., if unit is 'paragraph,' distance of zero means "same paragraph".) Distance is never negative.
- Relation: LessThan, LessThanOrEqual, Equal, GreaterThanOrEqual, GreaterThan, or NotEqual.
- Unit: Character, Word, Sentence, Paragraph, Section, Chapter, Document, Element, Subelement, ElementType, Byte, or a privately defined unit.
- Ordered flag: if set, the test is for "right" proximity only (the left ordinal must not exceed the right ordinal and Distance is compared with the difference between the right and left ordinals); otherwise, the test is for "right" or "left" proximity. (Distance is compared with the absolute value of the difference between the left and right ordinals.)
- Exclusion flag: if set, "not" is to be applied to the operation (for example if the test with Exclusion flag 'off' is "'cat' within 5 words of 'hat'," then the same test with Exclusion flag 'on' is "'cat' not within 5 words of 'hat'").

Example: suppose A and B respectively specify "personal name = 'McGraw,J.' " and "personal name= 'Stengel, C.' ," and:

- Distance is 0,
- Relation is 'equal,'
- Proximity-unit is 'paragraph',

- Ordered flag is ’false’,
- Exclusion flag is ’false’.

Then the result is the set of records in which both of the personal names occur within the same paragraph. Using the same example, if the Exclusion flag is set to ’true,’ the result is the set of records in which the two personal names never both occur within the same paragraph.

If the Ordered flag is set to ’true’ (and Exclusion flag to ’false’) then the result is the set of records in which the personal name ’McGraw, J.’ occurs within the same paragraph as, but before, the personal name ’Stengel, C.’.

If distance is instead 1 (’ordered’ and ’exclusion’ flag ’false’) the result is the set of records in which the two personal names occur in adjacent paragraphs. If, in addition, Relation-type is ’less-than-or-equal’ the result is the set of records in which the two names occur within the same or adjacent paragraphs.

3.7.2.2 Extended Result Set Model for Proximity

In the extended result set model for proximity, the target maintains positional information, in the form of one or more position vectors, associated with each record represented by the result set, which may be used in a proximity operation as a surrogate for the search that created the result set.

Example:

Let R1 and R2 be result sets produced by type-1 query searches on the terms ’cat’ and ’hat’. In the extended result set model for proximity, the target maintains sufficient information associated with each entry in R1 and with each entry in R2 so that the proximity operation "R1 near R2" would be a result set equivalent to the result set produced by the proximity operation "cat near hat" ("near" is used here informally to refer to a proximity test).

The manner in which the target maintains this information is not prescribed by the standard. Appendix 13 ERS (non-normative) provides examples.

3.7.3 Restriction and the Extended Result Set Model

The Restriction operand specifies a result-set-id and a set of attributes, and it represents the set of database records identified by the specified result set, restricted by the specified attributes.

Example:

Let R be the result set produced by a search on the term ’cat,’ representing three records:

1 where ’cat’ occurs in the title,
2 where ’cat’ occurs in the title and as an author, and
3 where ’cat’ occurs in the title, as an author, and subject.

Then "R restricted to ’author’" might produce the result set consisting of the entries 2 and 3 of R.

In the extended result set model for restriction, the target maintains information associated with each record represented by the result set, that may be used in the evaluation of a restriction operand as a surrogate for the search that created the result set. The manner in which the target maintains this information is not prescribed by the standard. Appendix 13 ERS (non-normative) provides examples.