ANSI/NISO

Z39.50-1994

Information Retrieval: Application Service Definition and Protocol Specification

Completed Preliminary Ballot Draft

August 1994

Status*: This is the completed preliminary ballot draft of Z39.50-1994 (earlier preliminary draft was incomplete), including phase-1 editing. This is submitted to NISO by the Z39.50 Maintenance Agency, in preparation for ballot. The actual ballot draft will differ in format and there will be minor editorial differences. The ballot period for Z39.50-1994 is scheduled to begin September 1.*

Contents

(Contents continued on reverse)

(Contents continued)

Appendices

1. Introduction

This standard, ANSI/NISO Z39.50-1994, *Information Retrieval 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 (Z39.50 service, section 3) and specifies the Information Retrieval application protocol (Z39.50 protocol, section 4). The service definition describes services which 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 use of and by 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, Z39.50-1994, 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), Version 2 of Z39.50 is assumed identical to version 1 of Z39.50; thus implementations which 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 Z39.58 -- *Common Command Language for Online Interactive Information Retrieval* 1991

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 -- *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 10163 -- *Documentation - Search and Retrieve Protocol Specification* 1991.
Library of Congress -- *USMARC Code List for Languages* 1989.
ISO -- *International Register of Coded Character Sets to be used with Escape Sequences* 1992.
IEEE 1003.2 -- *Portable Operating System Interface -- Part 2: Shell and Utilities* 1992.

2. Definitions

A-association	See "Application association".
Abstract database record	An abstract representation of the information in a database record; it 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.
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 which 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.
Aggregate present response	Segment requests (if any) together with the Present response, for a Present operation.
APDU	See "Application Protocol Data Unit".
Applied variant	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".
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 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.
Application Protocol	The rules governing the format and exchange of information between an origin and target.
ASN.1	Abstract Syntax Notation One, as specified in ISO 8824 and ISO 8825.

ARS See "Abstract record structure".

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: an attribute type and a value of that type.

Attribute list A set of attribute elements, and for each, 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 which 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 which may be included in a Present request to indicate the desired composition (elements and record syntax) of the retrieval records. It includes a schema name, element specification, and record syntax name.

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".

Data element See "Element".

Database A collection of information units; each unit is a collection of related information referred to as 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, 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.
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 which is 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 name	The name of an element specification format or an element set name.
Exceptional record size	The maximum size of the (single) 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 consist of the Search service. Alternatively, a facility might not consist of services, but instead 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 which 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, decscribed in section 3.3.3; within that section, a record is considered to be a string of bytes.)
Initiating request	A request that initiates an operation.
Intermediate fragment	A fragment which neither starts at the beginning nor ends at the end of a record. See "Fragment".
IR	Information Retrieval.
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, which 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. There may be multiple concurrent operations 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 which 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, 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, and/or used by the target, to represent retrieval records. For a complete definition, see 3.6.3.

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 entries, each consisting of a database name, a pointer to a record within the database, and possibly, unspecified information associated with each record, which may be used as a surrogate for the search that created the result set.

Result set record An idiomatic expression referring to the database record represented by a result set entry (see "Result set").

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

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

RPN Reverse Polish Notation.

RPN query A search query represented in RPN format.

Schema See "Database schema".

Segment A message which 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 primitive An abstract, implementation-independent representation of an interaction between the service-user and the service-provider. There are four types of service primitives: Request, Indication, Response and Confirmation.

Service-provider An abstraction of the totality of those entities (the origin and target) which 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 which 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 which 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.

Target The entity that accepts a Z-association.

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

Terminating response A response that terminates an operation.

Transfer syntax A syntax, paired with an abstract syntax, to form a record syntax.

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, appliedVariant, 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 An association 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 which 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 no status information is retained once a Z-association is terminated, except information which 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 <u>request</u> or a <u>response</u>. Services are defined to be <u>confirmed</u>, <u>non-confirmed</u>, or <u>conditionally-confirmed</u>.

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 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 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 Operations

There are eight <u>operation types</u> described within this standard: 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 <u>initiating request</u> and a response that terminates an operation is called a <u>terminating response</u>. 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 which can be specified in a search for database records (see 3.1.4), and one or more sets of elements which may be retrieved from a database record (see 3.1.5). An access point is a unique or non-unique key which 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 search 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 search 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 name, element specification name, and record syntax name.

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 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 Extended Services

One particular Z39.50 service is the Extended Services (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 completed; in fact the ES request may specify that the task should complete prior to the ES response). An ES task may have a lifetime beyond the Z-association.

Examples of extended services are:

- save a result set
- save a query
- schedule a periodic query
- Create an export specification
- Invoke an export specification
- Order an item
- Update a database

Each ES task is represented by a database record, called a task package, maintained in a special database by the target. This "extended services 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 if the task has started.

The origin, via an ES request, may 'create', 'delete', or 'modify' a task package. When it creates a task package, it specifies a user-id, to be associated with the package, as well as permissions for other user-ids. Permissions include 'create', 'delete', and 'modify', as well as 'present', and 'invoke'.

For example, when an origin creates a task package of type SaveResultSet a (persistent) result set is created, represented by the created task package at the target. When that package is retrieved (either in a the same or a subsequent Z-association), a copy of that persistent result set is made available to that Z-association as a Z39.50 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.

As another example: when a user creates a task package (i.e. the origin creates the task package on behalf of the user) of type SaveQuery, a persistent query is created, represented by the created task package. Another user may subsequently create a PeriodicQuerySchedule task package, which refers to the persistent query task package, only if that user has 'invoke' privilege for that persistent query. The periodicQuerySchedule task package includes an 'active' flag, indicating whether the periodic query is active. The user may subsequently 'modify' the task package (if the user has 'modify' privilege for the package) to turn the 'active' flag 'on' or 'off'. Similarly, a user may create an ExportSpecification (package) and another user (or the same user) may subsequently 'invoke' that ExportSpecification, by creating an InvokeExportSpecification package.

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

Section 3.2.1 through 3.2.11 describe the eleven facilities of the Information Retrieval service. Most consists 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.

(1) **Initialization Facility**

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

(2) **Search Facility**

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

(3) **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.

(4) **Result-set-delete Facility**

Delete Service: A confirmed service initiated by the origin to initiate a Delete operation.

(5) **Browse Facility**

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

(6) **Sort Facility**

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

(7) **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.

(8) **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.

(9) Explain Facility

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

(10) Extended Services Facility

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

(11) 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. The origin proposes values for initialization parameters in the Init request. 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 (optional)	
Options	x	x
Preferred-message-size	x	x
Exceptional-record-size	x	x
Result		x
Implementation-id	x (optional)	x (optional)
Implementation-name	x (optional)	x (optional)
Implementation-version	x (optional)	x (optional)
User-information-field	x (optional)	x (optional)
Other-information	x (optional)	x (optional)
Reference-id	x (optional)	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 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, what the value is. 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.

<u>search, present, delete, resource-report, scan, sort, and extended-services</u> - for each operation type, 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 effect, even though they are not defined in ANSI Z39.50-1992.

<u>trigger-resource-control</u> - The origin may propose to submit Trigger-resource-control requests; if so, the target indicates whether it will accept Trigger-resource-control requests. If 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', than 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) An indication by the target 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.

<u>level 1 and level 2 segmentation</u> - 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.

<u>concurrent-operations</u> - 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.

<u>named-result-sets</u> - 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'.

<u>resource-control and access-control</u> - 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 if 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 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 clarification.

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 either the origin or target service user 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 <u>result set</u>, 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. Five 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 later versions 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 which 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 which is 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); or
- 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 data-base 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 which 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 search 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 <u>response record</u> 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 than 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 response 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 dissambiguate certain 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 format or content of that information. User Information format SearchResponse-1 is defined in appendix 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 effect.

3.2.3.1.3 Result-set-id The origin indicates a result set, created by a previous Search request during this Z-association, from which records are to be retrieved.

3.2.3.1.4 Element-set-names 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 Max-segment-count may be included when segmentation is in effect. Max-segment-size and/or Max-record-size may be included when level 2 segmentation is in effect. The use of these parameters is detailed in 3.3. They may be used only when version 3 is in force.

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 which case, the positions will be ascending but may have gaps, but those gaps 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 response 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) This service may be used only when version 3 is in force.
(2) 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.
(3) 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 both response records and 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, which are 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 which 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:
    - o 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.
    - o 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:

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.

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 if 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).

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:

If concurrent operations is in effect, following initialization, the origin must be prepared at any time during the association, whether or not there are active operations, to receive an Access-control request pertaining to the Z-association, respond with an Access-control response, then later receive another Access-control request, etc.

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.

If the origin fails to respond correctly to the challenge, the target might decide to terminate one or more operations but leave open the Z-association. In that case, the target's terminating response to any such operations may indicate that the operation was terminated due to 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 ACC of this standard. 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.