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-report request, which includes 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:
  - o 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.
  - o 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 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 Suspended-flag is off, the partial results available situation may change since 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 pertained 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 Identifies a resource report format that the origin 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 operation.

> 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 wants 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 which 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	-	Access 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.

3.2.6.3.7 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 created, whose content is the sorted results. If Sorted-result-set is not the name of an existing result set: 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 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 a diagnostic 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 either the origin or target for additional information, not specified by the standard.

3.2.2.7.1.8 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 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 (optional)	x (if applicable)
Number-of-entries	x	x
Position-in-response	x (optional)	x (optional)
Scan-status		x
Entries		x (optional)
Other-information	x (optional)	x (optional)
Reference-id	x (optional)	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 would consist of a single attribute (from the bib-1 attribute set definition) 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. 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 (reason specified by Scan-status), or
- one or more non-surrogate diagnostic records indicating that the operation cannot be processed, and why it cannot be processed.

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

- 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 either 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 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 and/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. 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 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 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 services. 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 any.

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, then if the specified task has not been acted on, it will not be started. If the task is active, the target will 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 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 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 either 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.

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 are defined and registered in Appendix ATR. The task package structures are defined and registered in Appendix 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 items, each containing a user Id and a sequence of allowed operations from the following set:

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

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 is no longer meaningful.

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 ATR (which defines a set of Use attributes and imports specific bib-1 attributes); and
- support the Explain syntax, which is defined and registered in Appendix 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 bib-1 relation attribute 'equal', position attribute 'any position in field', and structure attribute 'key'. Origins should not 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 Use attribute = 'ExplainCategory' and with a value for each key for that category should result in 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 Section 3.2.10.3 each category is described in detail.

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 sets, including the attributes supported within the set.	attribute set oid
ScanInfo	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
Element Details	An element (for a particular record syntax, defined by a particular schema).	databaseName, schema oid, recordSyntax oid
SortDetails	A sort specification for a database.	database name
Processing	Processing directions for a database.	database name, processing-context
CategoryList	Explain categories that the target supports.	(no key)

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='ScanInfo' 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.
- To search for the description of elements defined for a record syntax in a specific schema, for a particular database, use ExplainCategory='ElementDetails' 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 the USMARC Code List for Languages). 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 a term of ’DatabaseInfo’, in combination with the Availability attribute with a term of ’yes’.
- To locate the databases provided by a specific supplier, use the ExplainCategory attribute with a term of ’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 a term of ’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 a term of ’DatabaseInfo’, in combination with the Proprietary attribute with a term of ’no’.
- to locate databases that have no user fee, use the ExplainCategory attribute with a term of ’DatabaseInfo’, in combination with the UserFee attribute with a term of ’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.

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.

- Which query-types are supported.
- Whether the proximity operator is supported, and if so, what units.
- Internet address and Port number.
- OSI addresses.
- 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.)
  - o Diagnostic sets supported.
  - o Resource reports supported.
  - o Access challenges supported.
  - o Attribute sets supported.
  - o Schemas supported.
  - o Record syntaxes 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).
- 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.

Non-brief elements:

- 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.
- Any 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.
- Supported query operators.
- Whether proximity operations are supported.
- Whether the origin should expect an access control challenge if this database is used.
- Text describing access control for this database, in human readable text.

- Costing information, in both machine readable format, and in human readable text, for connect, present, and search.
- Attribute set Ids supported by this database.
- Schemas defined for this database.
- Record syntaxes supported by this database.
- Element set names supported, with names and descriptions given in human readable text.
- Diagnostic sets supported by this database.
- Resource reports supported by this database.
- Access challenges supported by this database.
- Proximity parameters supported.

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:
  - o The name of the element.
  - o The tag assigned to the element.
  - o A description of the element.

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:

- A list of all attributes, and the following "Attribute Details" per attribute:
  - o Attribute name.
  - o Attribute description in human readable text.
  - o Attribute type.
  - o Attribute value.

3.2.10.3.7 Scan-Info Descriptive information about term-lists for scanning. 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:
  - o Name of the term-list.
  - o An indication of how expensive it is to search, using the associated attributes.
  - o Whether the term-list is scanable.

(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:
  - o Whether it is a private extended service.
  - o Whether restrictions apply.
  - o Whether a fee applies.
  - o Whether the service is available.
  - o 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 oid of the attribute set, and for each attribute within the set:

  - o The attribute type.
  - o A default value which applies if the attribute is omitted, and a description of default behavior in human readable form.
  - o 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.

Brief elements:

- A name for the term-list.

Non-brief elements:

- A description.
- Combination of Attributes used by Scan to access the term-list.
- 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.
- 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 by the Element-Details category.

3.2.10.3.12 Element-Details Descriptive information about an element of a record. Note that an element is relative to a database schema. There is one such Explain record for each element for each database for each schema for each record syntax.

Brief elements:

- The database to which this record pertains.
- Schema in which this element is defined.
- record syntax for which this Explain record pertains.

Non-brief elements:

- A description of this element, in human readable text.
- The tag of the element, if any.
- A list of schema element 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:
  - o A description.
  - o If the key is a record element, a specification of the element.
  - o If the key is an attribute combination, a specification of that combination.
  - o The type of key: character, numeric, structured.
  - o Whether the key is case sensitive.

3.2.10.3.14 Processing-Info Description of how the target believes the data should be processed by the origin for presentation to the user. There is one Explain record of this type for each database, for each processing context for which the target offers processing information.

Brief elements:

- Full name of the database to which this record pertains.
- The context for which this processing information is pertinent, i.e. access, search, retrieval, record presentation or record handling.
- A name for this processing information, in human readable form.
- An object identifier, if externally defined instructions (below) are included. Thus origin can retrieve the brief record and determine if it can handle instructions when retrieving the full record.

Non-brief elements:

- Instructions: one or both of the following:
  - o A description of the processing information, in human readable form.
  - o Externally defined instructions.

3.2.10.3.15 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 items 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 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 (optional)	x (optional)	
Resource-report-format	x (optional)		Origin only
Resource-report	x (optional)	x (optional)	Target only
Other-information	x (optional)	x (optional)	

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, the last of the above listed statuses, "response to Close request" may (but need not) be used to signal that the message was indeed issued as a response.

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 targets 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: it may unilaterally include a resource report.

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

3.3 Message/Record Size and Segmentation

A "segment" is 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.

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).

There are two levels of segmentation, level 1 and level 2, 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 Which Apply 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.

<u>Illustration</u>

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.

<u>Exception</u>

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, but to also provide 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 parameter Max-segment-count, which may optionally be supplied on the Present request, when segmentation is in effect.

If the value of Max-segment-count 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:
   - o Fit records P through N in the segment.
   - o Go to step 3.

   Otherwise,
   - o 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.
   - o If Max-segment-count is reached, go to step 3.
   - o Send a the segment as a Segment request.
   - o Set P=Q+1.
   - o 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.

<u>Illustration</u>

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.

(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:
- o a segment request including records 1 through 4,
- o a segment request including records 5 through 9,
- o a Present response including record 10.

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:
- o a segment request including records 1 through 4,
- o a Present response including records 5 through 9.

Present-status would be Partial-2.

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 defined to be 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 which starts at the beginning of a record. An intermediate fragment is a fragment which neither starts at the beginning nor ends at the end of a record. A final fragment is a fragment which 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 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 number of segments in the aggregate present response will not exceed max-record-size times the number of records requested, divided by Max-segment-size.

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 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:"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 such 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) such 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 the 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 such 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;
- 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 there are no semantics 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 that there is 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.

There are no restrictions on the re-use or management of Reference-ids by the origin (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). 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 names.

An element specification name is the name 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 names 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 name (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" and which when applied to an abstract database record results in the same record.

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 which 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 the 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 Type-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	::=	AttributeSet+Term | ResultSetId | Restriction
Restriction	::=	resultSetId+attributeSet
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 includes 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, AttributeSet+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 items 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) attributeSet+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 result set identified by ResultSetId.

(c) Restriction operand (ResultSetId+attributeSet): 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.

- If the operator is AND, let S be the intersection of S1 and S2.
- If the operator is OR, let S be the union of S1 and S2.
- If the operator is AND-NOT, let S be the set of elements in S1 which are not in S2.

- If the operator is Prox:
  - o If both operands are of form (a) let S be 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.
  - o 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 then 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 item 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, GreaterOrEqual, Greater, 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 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 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 ERS (non-normative) provides examples.