Network Working Group   Jacob Palme
Internet Draft   Stockholm University/KTH
draft-ietf-palme-select-00.txt   Johan Kaers
Intended-for: Proposed standard   Starlab
Expires: December 2000   June 2000

The SELECT Protocol for Rating and Filtering

Status of this Document

Abstract

The SELECT protocol allows Internet users to supply their ratings of Internet documents, and to use ratings provided by other users to filter and select what to read In particular, SELECT supports so-called collaborative filtering. By this is meant that the filtering and selection for a particular user is made based on ratings provided by special groups of raters, such as peer groups, people with similar values, interests and expertise as the person for whom the selecting and filtering is done.

The SELECT functionality is downwards compatible with PICS [PICS 1, PICS 2], but a major difference is that while PICS is mainly oriented towards keeping out unsuitable information from children (blackballing), SELECT is mainly oriented towards helping people find the best and most valuable information for them on the Internet (goldballing). A syntactical difference from PICS is that the encodings in SELECT are using the XML encoding format.

More information

More information and links to the most recent versions of this document can be found at http://dsv.su.se/jpalme/ietf/selprot.html. A mailing list will be started in the middle of June 2000, for information on how to subscribe see the above URL.

Table of Contents

1. Terminology

Term	Description
----	-----------
Aggregate rating	A rating, which is computed based on one or more atomic ratings, combined in some way. A SELECT server may provide several different aggregate ratings, computed in different ways, for example based on only non-anonymous ratings, based on only ratings by certain experts or members of a specific peer group. Also different aggregation methods can be used, such as average, median or lower quartile. (Using the lower quartile will favour controversial documents, which may sometimes be desirable.)
Anonymous rating	A rating, where the rater is not idenfied. See chapter 4.
Atomic rating	A rating provided by one user on one document. Several atomic ratings of the same document can be combined to produce an aggregate rating. See chapter 7.2.
Collaborative rating	A rating provider for a user, based on ratings made by other users in a peer group, which has shown itself to have the same rating values as the user getting the rating. '
ML	Machine Learning: Technology where a computer program learns by itself by observation of reality, for example by observation of human behaviour.
NLP	Natural Language Processing: Processing of natural language text with programs, which can in some way analyze it, for example derive genre information from the text style.
Non-anonymous rating	A rating, where the rater has to log in and identify itself before being allowed to provide a non-anonymous rating. See chapter 4.
Rating	Ratings are collections of descriptors of resources, which can be used as a basis for filtering.

Term	Description
----	-----------
User	An agent providing ratings or using SELECT services. Can represent a user, but ratings may also be provided by other ways than direct user input, such as observation of user behaviour or linguistic analysis of documents.

2. Definitions

Any occurence of "http://select/" in this document should in actual usage be replaced by the URL of a particular SELECT service.

3. Protocol elements summary table

The SELECT protocols allow a SELECT server to keep a data base of ratings made my many different people on a resource, lika a web page. This data base can be used to find the aggregate ratings on resources and to search, using the aggregate rating as search criterium. Every server decides which and how many rating categories it provides.

In addition to manually added ratings, the SELECT data base can also store ratings made by a machine, such as an NLP engine, and by automatic observation of user behaviour.

The SELECT protocol can be used by Internet User Agents to submit ratings, get ratings, search and filter for a user. Such User Agents can be user-oriented servers, news servers and clients, web browsers, and plug-ins and client-side and server-side proxies.

The SELECT protocol can also be used by autonomous agents, which can perform services like NLP rating, finding news of special interest to a particular user and e-mailing the user with this, etc.

Name	**Task**	**Client(s)**
----	**----**	**---------**
Get service-description-list	Find out which rating services are handled by this server.	Input reader ratings / Another SELECT version 1.0 server / A filtering process, a user or manager
Get-service-de	Find out which rating	Input reader ratings /

Name **----**	**Task** **----**	**Client(s)** **---------**
scription	descriptors are handled by this rating service.	Another SELECT version 1.0 server / A filtering process, a user or manager
Send-rating	Send a new atomic rating on a resource.	Input reader ratings / Automatic rating agents
Set-profile	Self-register a rater with a server, as well as registering someone else as a rater for a closed server, or modifying the profile of an existing user.	Input reader ratings, a user or manager
Get-profile	Get the profile of another user, subject to access controls.	Input reader ratings, a user or manager
Login	Establish credentials for a user.	All of the above
Logout	Waive credentials for a user.	All of the above
Get-atomic-rat ings	Get the ratings made by one or more named users on one or more resources.	All of the above
Simple-Search	Make a search for rated resources, HTML search query form.	Rating search client, a user.
Advanced-Searc h	Make a search for resources, XML query form.	Rating search client, NLP module (to find items which need NLP ratings)

Name ----	**Task** ----	**Client(s)** ---------
Evaluate	Get the ratings for a list of resources.	Rating search client, news client, news server, a user.
Exchange-ratings-data	Mirror ratings data between two SELECT version 1.0 servers.	One SELECT version 1.0 server.

4. Handling of anonymous ratings

Ratings can be either identified or anonymous.

All SELECT services may not allow anonymous ratings.

Anonymous ratings are fully anonymous, no raterid of any kind is specified.

Anonymous ratings are sent to a different URL (see chapter 6), containing /id/, than the URL for non-anonymous ratings. For anonymous ratings, the "raterid" has the special value "anonymous".

When combining atomic ratings to aggregate ratings, different weight may be given to identified, anonymous ratings, including the weight zero to anonymous ratings.

When retrieving atomic ratings, you will get identified ratings only for yourself. Other ratings are not returned or are returned only unidentifiable format.

5. Style sheet information in XML encodings

The XML encodings produced by SELECT agents may contain style sheet information. An agent which does not use this information, should ignore it. Such an agent must be capable of receiving and ignoring style sheet information, but need not do any other processing of style sheet information.

Such style sheet information may be:

(a) A style sheet reference in the processing instruction head of an XML document.

(b) A style sheet reference in the DTD file (not valid today, August 1999, but may become valid in the future).

Example: The following two XML data are semantically equal:

Version 1: ---------	**Version 2:** ---------
<?xml version="1.0"?> <!DOCTYPE send-rating-response SYSTEM "http://select/v1.0/send-rating-response.dtd"> <send-rating-response accepted="false" refuse-reason="accesscontrol"/>	<?xml version="1.0"?> <?xml-stylesheet href="mystyle.css" type="text/css"> <!DOCTYPE send-rating-response SYSTEM "http://select/v1.0/send-rating-response.dtd"> <send-rating-response accepted="false" refuse-reason="accesscontrol" class="error"/>

6. Submission points

Below are shown the entry points for access to the SELECT general service. For a specialised service, the word "general" below should be replaced by the subdirectory for that service.

Operation	**Service**	**Submission point**
Get-Service-Description-List	All services	http://select/v1.0/select-service-descriptions.xml
Get-Service-Description	General service	http://select/general/get-services.xml
Send-Rating	General service	http://select/v1.0/general/id/input-ratings Note: For use by ratings supplied by identified raters.

Send-Rating	General service	http://select/v1.0/general/id/input-ratings Note: For use for ratings supplied by identified raters.
Send-Rating	General service	http://ano.select/v1.0/general/input-ratings http://id.select/v1.0/general/input-ratings Note: ano.select is used to record anonymous ratings, id.select to record non-anonymous ratings. The different domain names are needed to keep the cookies different.
Set-Profile	General service	http://select/v1.0/general/id/profiles
Get-Profile	General service	http://select/v1.0/general/id/profiles
Login	General service	http://select/v1.0/general/id/login
Logout	General service	http://select/v1.0/general/id/logout
Get-Atomic-Ratings	General service	http://select/v1.0/general/id/evaluator
Simple-Search	General service	http://select//v1.0/general/simple-search
Advanced-Search	General service	http://select/search
Evaluate	General service	http://select/evaluator

7. Protocol elements full specifications

7.1 Validation of XML Encodings

The DTDs and XML code in this specification has been validated using the XML validation service at http://www.stg.brown.edu/cgi-bin/xmlvalid/xmlvalid.pl.

7.2 The DTD for an atomic rating

Explanation **-----------**	**Format of information sent** **--------------------------** (http://select/v1.0/atomic-rating.dtd)
	<!ELEMENT atomic-rating (rating-value+)>
Start of attribute list for atomic-rating.	<!ATTLIST atomic-rating
Identification of the rater. For a pseudonymous rating, the pseudonym is specified, for an identified user the raterid is supplied. For an anonymous rating, the special name "anonymous" is entered. Note: when retrieving ratings made by other people than yourself, the raterid is given the special value "suppressed". You will then get ratings with the raterid changed to "suppressed" and with only the year in the date field. For aggregate ratings, the raterid-or-pseudonym has the special value "derived".	raterid-or-pseudonym CDATA #REQUIRED
IP adress of machine that generates the rating.	rated-from-host CDATA #REQUIRED
Identification of the software which formatted this rating in the format of an URL of a page describing this rating-engine.	rating-engine CDATA #REQUIRED
The URI of the rated resource	location CDATA #REQUIRED

Explanation -----------	**Format of information sent** -------------------------- (http://select/v1.0/atomic-rating.dtd)
The date of the rating. If no date is specified, the server will give the record the current date when storing it. When returning ratings with the Get-Atomic-Ratings operation, ratings for other people than yourself are supplied with only year, not month or day or time-of-day. For aggregate ratings, this date has as value the last time its value was re-computed.	rating-date CDATA #IMPLIED
The rater-competence is an enumerated XML attribute, which can only take specified values. The default value, if no tag is specified, is "un-nown". "multiple" is used for aggregate ratings based on multiple user's atomic ratings.	rater-competence (author | expert | user | pseudonymous | anonymous | un-known | multiple) 'un-known'
The verification info for this rater, default is "anonymous" "multiple" is used for aggregate ratings based on multiple user's atomic ratings.	rater-trust (signed | registered | pseudonymous | anonymous | multiple) 'anonymous'
Type of agent producing this rating value.	rater-type (computed | observed | manual) 'manual'
Rating is limited to this context.	context (general | business | leisure | shopping | research | politics | all | not-available) 'not-available'

Explanation **-----------**	**Format of information sent** **--------------------------** (http://select/v1.0/atomic-rating.dtd)
The sender can give a globally unique message-id to the rating sent. If the sender does not give such an ID, then the recipient will assign such an ID when storing the rating. Note: This is not the Message-ID of the rated resource, it is the ID of this rating.	message-id CDATA #IMPLIED
End of the list of XML attributes.	>
The value for one rating descriptor.	<!ELEMENT rating-value EMPTY>
	<!ATTLIST rating-value
The transmit-as or short-name of the rating category. Note: Only descriptors which are defined for the rating service, to which the connection is made, are allowed!	type CDATA #REQUIRED
The value of the rating in the format specified for this rating descriptor. Note: If a user specifies more than one keywords for a resource, then each keyword is sent as a separate rating-value.	Value CDATA #REQUIRED
End of the list of XML attributes.	>

7.3 Get-Service-Description-List (XML)

The Get service-description-list operation retrieves a list of SELECT version 1.0 service descriptions and their URIs, but does not retrieve the actual service descriptions. This will not necessarily be a list of all SELECT services over the world, it may usually be a list of SELECT services on this particular host, or a list of services recommended by the manager of this host.

7.3.1 Query format (get service description-list):

An HTTP GET operation is performed on a URI established to return SELECT version 1.0 service descriptions.

Example:

This description can be requested from at:
http://select/v1.0/select-service-descriptions

Explanation **-----------**	**Information sent** **----------------**
Get the file named "sel-1/select-service-descriptions". Preferred language is in English, second choice Italian	GET /v1.0/select-service-descriptions HTTP/1.1
From the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

7.3.2 Response format (get service description-list):

Explanation **-----------**	**Information sent** **----------------**
Standard reply header	HTTP ...

Explanation -----------	Information sent ----------------
	<?xml version="1.0"?> <!DOCTYPE send-rating SYSTEM "services-list.dtd"> <services-list> <services-list-item id = "test" URI = "http://samson.aszi.sztaki.hu/SELECT" server = "samson.aszi.sztaki.hu" maintainer = "micsik@sztaki.hu" /> <description language = "en" text = "SELECT service"> </description> </services-list-item> </services-list>
DTD of replied message	<!ELEMENT services-list-item (description+)> <!ATTLIST services-list-item id CDATA #REQUIRED URI CDATA #REQUIRED Server CDATA #REQUIRED Maintainer CDATA #REQUIRED > <!ELEMENT description EMPTY> <!ATTLIST description language CDATA #REQUIRED text CDATA #REQUIRED >

7.4 Get-Service-Description

Summary: The Get service-description operation will query a SELECT version 1.0 server to get a description of some services. The main components of this description is a list of descriptors and scales used by this service.

Access control: None.

Input data: The names of the services.

Output data: A description of the service, and a list of the descriptors supported for ratings in that group.

Base protocol: HTTP combined with XML.

7.4.1 Query format (get service-description):

An HTTP GET operation is performed on a certainURI.

Example (get service-description):

This example retrieves the service description at the URI:
http://select/v1.0/get-service-descriptions

Explanation -----------	**Information sent** ----------------
Get the file named "general/select-service-description" which contains a description of the SELECT version 1.0 general service. The SELECT version 1.0 general service is a service available to everyone, as different for service for special user groups.	GET /v1.0/get-service-descriptions HTTP/1.1
Get the file from the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"
Request of SELECT test service	<?xml version="1.0"?> <!DOCTYPE get-services SYSTEM "get-services.dtd"> <get-services> <service-name name = "test"/> <service-name name = "iscn"/> </get-services>
DTD of message	<?xml version="1.0" encoding="UTF-8" ?> <!ELEMENT get-services (service-name+)> <!ELEMENT service-name EMPTY> <!ATTLIST service-name name CDATA #REQUIRED >

7.4.2 Response format (get-service-description):

The response is an XML [XML1], [XML2] resource, containing a SELECT version 1.0 service description. The XML Resource Type Declaration for this XML page is described in chapter 0

The SELECT general service description.

7.5 Send-Rating

Summary: The send-rating operation will send one or more ratings to a SELECT version 1.0 server. This operation can be used both for explicit ratings provided by users, for implicit ratings derived by observing user behaviour, and for ratings derived through automatic analysis of documents using NLP methods.

Access control: If the rater is not identified by a cookie (created by a login operation), then either this rating will be handled as anonymous or the user will be instructed to login first, or to send the ratings to the separate entry-point for anonymous ratings. Some SELECT servers may not accept anonymous ratings.

Input data: Information about the rated resource, the rater and the rating values.

Output data: Acceptance or rejection.

Base protocol: XML transported through HTTP.

7.5.1 Transmit-Format (send-rating):

A HTTP POST operation, with the content the XML-formatted rating.

The send-rating is an HTTP POST operation, whose body is an XML resource containing the rating. Below is a POST sent to the URI http://select/v1.0/general/input-ratings

Explanation -----------	**Information sent** ----------------
Connect to the SELECT server. The URI used identifies the rating service, to which this rating is sent.	POST /v1.0/general/id/input-ratings HTTP/1.1

Explanation **-----------**	**Information sent** **----------------**
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
The format of the query is XML.	Content-Type: Application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

The body of the query is an XML [XML1], [XML2] resource. The XML Resource Type Declaration for this XML resource is as follows. Note that no Rater-ID is included, because this ID can be derived from the Cookie. And no rating-service-description is referred to, because the URI, to which this rating is sent, implies a particular rating-service.

Explanation **-----------**	**Format of information sent** **---------------------------** (http://select/v1.0/send-rating.dtd)
Reference to data structure defined in a separate DTD file. Further information, see section 7.2	<!ENTITY % atomic-rating SYSTEM "http://select/v1.0/atomic-rating.dtd">
A list of ratings are sent.	<!ELEMENT send-rating (atomic-rating+)>
Import DTD from separate DTD file atomic-rating.dtd.	%atomic-rating;

Example (send-rating):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/send-rating.xml)
HTTP header	POST /v1.0/general/id/input-ratings HTTP/1.1 Host: select Accept: application/xml Content-Type: Application/xml Cookie: session="1234567890123456"
A blank line to mark the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE send-rating SYSTEM "http://select/v1.0/send-rating.dtd">
	<send-rating>
Start with information about the resource rated and about the rater.	<atomic-rating raterid-or-pseudonym="jpalme@dsv.su.se" rating-engine="http://select/proxy-1" location="http://www.body.com/eyes" rating-date="31 Jul 1999" rater-competence="user" rater-type="manual" rater-trust="registered" message-id="990815113350*jpalme@dsv.su.se">
First rating descriptor	<rating-value type="select-reader-interest-rating" value="good"/>
Second rating descriptor, note that decimal values are allowed	<rating-value type="select-reader-quality-rating" value="1.5"/>
Third rating descriptor	<rating-value type="adult" value="false"/>

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/send-rating.xml)

Fourth rating descriptor	<rating-value
	  type="context"
	  value="leisure"/>

End of data	</atomic-rating></send-rating>

7.5.2 Response format (send-rating-response):

The response is an XML [XML1], [XML2] resource. The XML Resource Type Declaration for this XML resource is:

Explanation **-----------**	**Format of information sent** **--------------------------** (http://select/v1.0/send-rating-response.dtd)
The evaluations are returned, one rating service at a time.	<!ELEMENT send-rating-response (rating-value*)>
Whether all the rating labels were accepted, or some of them, or none of them.	<!ATTLIST send-rating-response accepted (all | some | none) 'all'
ID of the set-rating. If no message-id was given in the set-rating operation, this ID will tell the client which ID the set-rating operation got assigned by the server.	Message-id CDATA #REQUIRED
If rating was rejected, explanation why. See chapter 7.5.2.1 Refusal reasons for the send-rating operation.	Refuse-reason (accepted | bad-syntax | Missing-info | unknown-descriptors | wrongtype | Wrong-competence | wrong-trust | wrong-rater-type | Wrong-context | access-control | other | Wrong-type) 'accepted'
End of XML attribute list	>
If only some of the rating values were rejected, this element is used to list the rejected rating values.	<!ELEMENT rating-value EMPTY>
	<!ATTLIST rating-value
See send-rating operation.	type CDATA #REQUIRED
See send-rating operation	value CDATA #REQUIRED

Explanation **-----------**	**Format of information sent** -------------------------- (http://select/v1.0/send-rating-response.dtd)
End of XML attribute list	>

7.5.2.1 Refusal reasons for the send-rating operation

The following refusal reasons may be used in rejecting a send-rating operation by a SELECT server:

Refuse-reason **-------------**	**Explanation** **-----------**
none	Operation was not rejected.
bad-syntax	Wrong syntax of HTTP header or XML data sent.
missing-info	Mandatory-information missing from sent data.
unknown-descriptors	Trying to store a rating for a descriptor not supported by this server.
wrongtype	Wrong type of a descriptor value, for example text for a descriptor which must have a numerical value.
wrong-competence	This rater is not allowed to send ratings with this competence to this service.
wrong-trust	This rater is not allowed to send ratings with this trust to this service.
wrong-rater-type	This service does not accept ratings of this rater-type from this user.
wrong-context	This service does not accept ratings with this context from this user.
access-control	This user is not allowed to send ratings to this service. (There are no operations in this specification to give people access rights. Some SELECT services may want to give only certain people the right to perform various operations, such as send ratings. How to do this is not described in this specification.)

Refuse-reason -------------	Explanation -----------
not-logged-in	The user performed an operation which requires login, but was not logged in.
other	Other errors.

Example 1 (positive send-rating response):

Explanation -----------	Information sent ---------------- (http://select/v1.0/send-rating-1.xml)
HTTP response header	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header	
Identifies that this is in XML format	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE send-rating-response SYSTEM "http://select/v1.0/send-rating-response.dtd" >
Accepted is default.	<send-rating-response message-id="990815113350*jpalme@dsv.su.se"/>

Example 2 (negative send-rating response):

Explanation **-----------**	**Information sent** **----------------** (http://select/v1.0/send-rating-2.dtd)
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE send-rating-response SYSTEM "http://select/v1.0/send-rating-response.dtd" >
All ratings were not accepted.	<send-rating-response accepted="some" refuse-reason="wrong-context" message-id="990815113350*jpalme@dsv.su.se">
Rating-label rejected, this server does not accept ratings in the leisure context.	<rating-value type="context" value="leisure"/>
End of send-rating-response.	</send-rating-response>

7.6 Set-Profile

Summary: The set-profile operation can be used for a rater to register him/herself (for services which allow this) and can be used by administrators to register raters (for services which do not allow self-registration). It can also be used to modify existing registrations.

Issues: The format of interest-profile is not specified. The format of reward-account is not specified.

Access control: The profile of a person is not modifiable by other people, only by that person him/herself, or an agent for that person, or certain certified SELECT processes, who will not divulge the profile to other people. A SELECT administrator may also usurp super-user privileges and perform this operation on anyone.

Input data: User identification and some profile attributes to be set or changed.

Output data: Accepted or rejected.

Base protocol: XML

7.6.1 Transmit format (set-profile):

The set-profile operation is an HTTP POST operation, whose body is an XML resource containing the profile, sent to the profiles cgi-script in the server for this particular rating service. Example: "http://select/v1.0/general/id/profiles/".

Note that a user, who is registered in more than one rating service, has a separate profile and a separate cookie for each of them.

Explanation -----------	**Information sent** ----------------
Connect to the SELECT server	POST /v1.0/general/id/set-profile HTTP/1.1
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
The format of the query is XML.	Content-Type: Application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

The body of the operation is an XML [XML1], [XML2] resource. The XML Resource Type Declaration for this XML resource is:

Explanation -----------	**Format of information sent** -------------------------- (http://select/v1.0/set-profile.dtd)
Reference to data structure defined in a separate DTD file. Further information, see section 7.6.2 The XML DTD for the user profile.	<!ENTITY % profile SYSTEM "http://select/v1.0/profile.dtd">

The set-profile consists of a profile plus two attributes.	`<!ELEMENT set-profile (profile)>`
Start of XML attribute list.	`<!ATTLIST set-profile`
Is this a new registration of a not-yet-registered user?	`new (true | false) 'false'`
Whether you are setting the registration for yourself, or, since you are a superuser, for someone else.	`self (true | false ) 'true'`
	`>`
Profile is taken from the external ENTITY declared in the first row. Further information, see section section 7.6.2 The XML DTD for the user profile.	`%profile;`

Note: All attributes of a user profile are not settable for ordinary users (example: no-of-docs-rated). They should thus not be used when setting a profile.

Pseudonym should include the domain name of the SELECT server. Thus, if a user wants the pseudonym foobar, the user should request the pseudonym foobar@select when connecting to any of the SELECT servers at select. Note that this means that the same pseudonym is not allowed in more than one SELECT service, if all the services are on the same server. The SELECT server must check suggested pseudonyms in a data base which is common to all SELECT services on a particular host.

7.6.2 The XML DTD for the user profile

Profile is an XML [XML1], [XML2] resource. The XML Resource Type Declaration for profile:

Explanation **-----------**	**Format of information sent** **--------------------------** (http://select/v1.0/profile.dtd)
	`<!ELEMENT profile (language*, keyword-manual*, keyword-automatic*, interest-profile?, query-history*, reward-account*)>`

	<!ATTLIST profile
E-mail address of the rater. This can be omitted for a person who is only going to submit pseudonymous ratings. One of the two values raterid and pseudonym must be specified.	raterid CDATA #IMPLIED
A globally unique identification of the rater, from which the real person cannot be found except through the SELECT server data base. (Such lookups are forbidden except when needed to fight illegal or harmful usage.)	pseudonym CDATA #IMPLIED
Password is mandatory except that a new user can omit the password at registration, the server will then assign a password to that user and send it back.	password CDATA #IMPLIED
A question to answer for a user who has forgotten his/her password, example "What is my mother's maiden name".	remember-phrase-question CDATA #IMPLIED
The correct answer to this question.	remember-phrase-answer CDATA #IMPLIED
A non-unique, user-friendly name of this rater.	name CDATA #IMPLIED
Wanted maximum validity time of a session before time out, in seconds. Both the client and the server are responsible for not allowing further interactions before logout when the validity time has expired. This value is dependent on the setting, which the user makes on a "Remember my password" checkbox in the user profile settings user interface.	cookie-life-time CDATA #IMPLIED
Open key for signatures. May be mandatory, optional or not used, depending on SELECT service.	signature-open-key CDATA #IMPLIED

<table>
<tr><td>URL of certificate authority, with which the signature-open-key can be verified.</td><td>certificate-authority CDATA #IMPLIED</td></tr>
<tr><td rowspan="2">A user does not have to specify birthyear or gender. Some SELECT services may however require such specifications. Birthyear is the year AD (counted from the commonly assumed birthyear of Christ).</td><td>birthyear CDATA #IMPLIED</td></tr>
<tr><td>gender (male | female) #IMPLIED</td></tr>
<tr><td>Some users may not have permission to rate.</td><td>mayrate (true | false) 'true'</td></tr>
<tr><td>Whether other people can search for this user's name in the SELECT data base.</td><td>secret (true | false) 'false'</td></tr>
<tr><td>Latest date when this rater made any rating in this rating service.</td><td>latest-rating-date CDATA #IMPLIED</td></tr>
<tr><td>Number of docs rated by this rater in this rating service.</td><td>no-of-docs-rated CDATA #IMPLIED</td></tr>
<tr><td>Algorithm not yet defined for how to compute this.</td><td>average-ratings-given CDATA #IMPLIED</td></tr>
<tr><td>End of the list of XML attributes</td><td>></td></tr>
<tr><td>Language code of languages understood by this user in priority order. May be repeated once for every language. Language codes are taken from RFC 1766 and ISO 639.</td><td><!ELEMENT language (#PCDATA)></td></tr>
<tr><td>Keywords specified by this user to identify his/her interests.</td><td><!ELEMENT keyword-manual (#PCDATA)></td></tr>
<tr><td>Keywords automatically derived by observation of this user to identify his/her interests.</td><td><!ELEMENT keyword-automatic (#PCDATA)></td></tr>
</table>

This syntax is preliminary. We may assign a more complex syntax to this later on, with defined subelements and structure like a set of instructions in some filtering language.	<!ELEMENT interest-profile (#PCDATA)>
List of previous queries made by this user. May influence filtering procedure.	<!ELEMENT query-history (#PCDATA)>
To be defined.	<!ELEMENT reward-account (#PCDATA)>

Example (set-profile):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/set-profile.xml)
HTTP header.	POST /v1.0/general/id/set-profile HTTP/1.1 Host: select Accept: application/xml Content-Type: Application/xml Cookie: session="012345678901234354"
A blank line to mark the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE set-profile SYSTEM "http://select/v1.0/set-profile.dtd">
Setting the profile for someone else.	<set-profile self="false">
Start of the profile to be set.	<profile

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/set-profile.xml)
List of attributes and values.	raterid="jpalme@dsv.su.se" pseudonym="xavier-xantico" password="foobar" remember-phrase-question="mother's maiden name" remember-phrase-answer="von Vegesack" name="Jacob Palme" cookie-life-time="999999999" birthyear="1941" gender="male"
End of profile attributes	>
Embedded elements	<keyword-manual>standards</keyword-manual> <keyword-manual>computers </keyword-manual> <keyword-manual>fiction </keyword-manual> <keyword-automatic>psychiatry </keyword-automatic> <interest-profile> Do not filter away any document containing "IETF"</interest-profile>
End of set-profile	</profile></set-profile>

7.6.3 Response format (set-profile):

<table>
<tr><th>Explanation
-----------</th><th>Format of information sent

(http://select/v1.0/set-profile-response.dtd)</th></tr>
<tr><td>The evaluation are returned, one rating service at a time.</td><td><!ELEMENT set-profile-response</td></tr>
<tr><td>Whether all the new user settings were accepted, or some of them, or none of them.</td><td><!ATTLIST set-profile-response
accepted (all | some | none) 'all'</td></tr>
<tr><td>XML attributes for refuse-reason.</td><td>reason (cannot-set-for-yourself | cannot-set-for-other-user | raterid-or-pseudonym-in-use | invalid-session-id | not-logged-in | database-error | failure | success) 'failure'</td></tr>
</table>

Example 1 (positive set-profile response):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/set-profile-response-1.xml)
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE set-profile-response SYSTEM "http://select/v1.0/set-profile-response.dtd" >
The set-profile was accepted.	<set-profile-response accepted = "all" reason = "none" />

Example 2 (negative set-profile response):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/set-profile-response-2.xml)
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE set-profile-response SYSTEM "http://select/v1.0/set-profile-response.dtd" >
The set-profile was not fully accepted.	<set-profile-response accepted="none" reason="raterid-or-pseudonym-in-use" />

7.7 Get-Profile

Summary: The *get-profile* operation can be used to get the profile settings for a particular user in a particular SELECT server. It can be used to retrieve a profile, and then send in a modified profile using (to the extent this is allowed) using the *set-profile* operation. Filtering agents may use *get profile* to get information used in the filtering for a certain user. ML algorithms may automatically modify a user's profile, the profile may indicate limits on what ML algorithms may do to it. (Example: "ML may not filter out any articles in newsgroup X, since it is very important to me".)

Access control: The profile of a person is not accessible by other people, only by that person him/herself, or an agent for that person, or certain certified SELECT processes, who will not divulge the profile to other people. A SELECT administrator may also usurp super-user privileges and perform this operation on anyone.

Input data: Identification of the user or search-info for the user, whose profile is wanted.

Output data: The profile of this user, or a rejection error.

Base protocol: application/x-www-form-urlencoded for the request, XML for the response.

7.7.1 Query format (get-profile):

The *get-profile* operation is an HTTP GET operation, with an HTML form:


```
<html>
<head>
<title>SELECT Login</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF">
<h1><font>Get SELECT User
  Info </h1>
<form method="get" action="http://www.dsv.su.se/~jpalme/">
  <table border="0" cellpadding="5">
    <tr>
      <td rowspan="2">Fill
        in either an e-mail<br>
        address or a search string:</td>
      <td>
        <div align="right">The
          e-mail address<br>
          of the user:</div>
      </td>
      <td>
        <input type="text" name="e-mail-address"
               size="50" maxlength="80">
      </td>
    </tr>
    <tr>
      <td>
        <div align="right">Search
```

```
          string:</div>
      </td>
      <td>
        <input type="text" name="search-string"
               size="50" maxlength="80">
      </td>
    </tr>
    <tr>
      <td> </td>
      <td> </td>
      <td>
        <input type="submit" name="Get" value="Get user info">
         <font face="Verdana, Arial, Helvetica" size="2">
        Response format:
        <input type="radio" name="responseformat" value="html"
               checked>
        HTML
        <input type="radio" name="responseformat" value="xml">
        XML </td>
    </tr>
  </table>
   </form>
</body>
</html>
```

Example:
"http://select/v1.0/general/id/profiles?e-mail-address=&search-string=Donald+Duck&Get=Get+user+info&responseformat=html".

Example (get-profile):

Explanation -----------	Information sent ----------------
Connect to the SELECT server.	GET /v1.0/general/id/profiles?raterid=jpalme HTTP/1.1
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml

7.7.2 Response format (get-profile):

<table>
<tr><th>Explanation
-----------</th><th>Format of information sent

(http://select/v1.0/get-profile-response.dtd)</th></tr>
<tr><td>Reference to data structure defined in a separate DTD file. Further information, see section 0.</td><td><!ENTITY %profile SYSTEM
"http://select/v1.0/profile.dtd"></td></tr>
<tr><td>The evaluation are returned, one rating service at a time.</td><td><!ELEMENT get-profile-response (profile+ | error)></td></tr>
<tr><td></td><td><!ATTLIST get-profile-response</td></tr>
<tr><td>Partial means that some, but not all the requested data is returned.</td><td>success (full | partial | none) 'full'</td></tr>
<tr><td></td><td>></td></tr>
<tr><td>Profile is taken from the external ENTITY declared in the first row. Further information, see section 0. Profile may be incomplete, in case only some attributes are retrievable for this requestor (if you get profile for someone else than yourself).</td><td>%profile;</td></tr>
<tr><td></td><td><!ELEMENT error (#PCDATA)></td></tr>
<tr><td></td><td><!ATTLIST error</td></tr>
<tr><td>Reject reason, no default value.
Note: Data for secret users, whom the requestor are not allowed to see, are treated as non-existing. A search for such a user might thus return "not-found".</td><td>reason (not-logged-in | bad-syntax |
not-found | authorisation-failure |
other-reason)
#IMPLIED</td></tr>
<tr><td></td><td>></td></tr>
</table>

Example 1 (positive get-profile response):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/get-profile-response-1.xml)
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE get-profile-response SYSTEM "http://select/v1.0/get-profile-response.dtd">
The get-profile was accepted.	<get-profile-response>
Start of the profile to be set.	<profile
List of attributes and values. Note: Password is never returned.	raterid="jpalme@dsv.su.se" pseudonym="xavier-xantico" remember-phrase-question="mother's maiden name" remember-phrase-answer="von Vegesack" name="Jacob Palme" cookie-life-time="999999999" birthyear="1941" gender="male"
End of profile attributes	>
Embedded elements	<keyword-manual>standards</keyword-manual> <keyword-manual>computers </keyword-manual> <keyword-manual>fiction </keyword-manual> <keyword-automatic>psychiatry </keyword-automatic> <interest-profile> Do not filter away any document containing "IETF"</interest-profile>

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/get-profile-response-1.xml)
End of get-profile	</profile></get-profile-response>

Example 2 (negative get-profile response):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/get-profile-response-2.xml)
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 17 August 1999 12:22:46 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE get-profile-response SYSTEM "http://select/v1.0/get-profile-response.dtd">
The get-profile did not succeed.	<get-profile-response success="none">
	<error reason="not-logged-in">
	You cannot do this without first logging in.
	</error></get-profile-response>

7.8 Login

Summary: The login operation is used to identify a user, and cause a cookie value to be set, which allows this user to perform certain access-controlled operations during the validity time of this cookie.

Access control: E-mail address and password. May not be required for sending anonymous ratings.

Input data: User identification by either e-mail address or pseudonym combined with password or an IMAP authentication.

Output data: Acceptance or rejection.

Base protocol: application/x-www-form-urlencoded for the request, HTML or XML for the response.

7.8.1 Query format (login):

The same as if the user has filled in the following HTML form:


```
<html>
<head>
<title>SELECT Login</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF">
<h1>SELECT Login </h1>
<form method="get" action="http://www.dsv.su.se/~jpalme/">
  <table border="0" cellpadding="5">
    <tr>
      <td>
        <div align="right">Your e-mail address<br>or pseudonym:</div>
      </td><td>
        <input type="text" name="e-mail-address" size="50"
maxlength="80">
      </td>
    </tr>
    <tr>
      <td>
        <div align="right">Your password:</div>
      </td><td>
        <input type="password" name="password">
      </td>
    </tr>
    <tr>
      <td> </td>
```

```
      <td>
        <input type="submit" name="Submit" value="Login"> 
        Response format:
        <input type="radio" name="responseformat" value="html" checked>
        HTML
        <input type="radio" name="responseformat" value="xml">
        XML
      </td>
    </tr>
  </table>
   </form>
</body>
</html>
```

Example (login):

Explanation **-----------**	**Information sent** **----------------**
Connect to the SELECT server	GET /v1.0/general/id/login?e-mail-address=jpalme@dsv.su.se&password=select HTTP/1.1
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml

7.8.2 Response format (login):

Explanation **-----------**	**Format of information sent** **--------------------------** (http://select/v1.0/login.dtd)
Response to a login	<!ELEMENT login-response EMPTY>
If ok Session-id for the newly created session Rater-id of the newly logged in user	<!ATTLIST login-response accepted (ok | wrong_password |unknown_user | failed) 'failed' session-id CDATA #REQUIRED rater-id CDATA #REQUIRED
End of XML attribute list	>

Example (login response):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/login-response.xml)
HTTP header.	HTTP/1.1 200 OK Date: Sun, 25 Jul 1999 13:32:18 +0200 Server: Apache/1.2.4 Last-Modified: Sun, 25 Jul 1999 13:32:18 +0200 ETag: "437e5-98-3531f2e3" Content-Length: 152 Accept-Ranges: bytes Connection: close Content-Type: application/xml
Set the cookie.	Set-cookie: session="1234567890123456";Domain="select";Path="/v1.0/general/id/"
A blank line to mark the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE login-response SYSTEM "http://select/v1.0/login-response.dtd">
Start and end of login-response for a rejected login.	<login-response accepted="true" session-id="1234567890123456" rater-id = "jpalme" />

7.9 Logout

Summary: The logout operation removes the cookie, which gave the user privileges to perform certain commands in logged-in state.

7.9.1 Query format (logout):

The same as if a user clicks on an HTML link:

<A HREF="http://select/v1.0/general/id/logout;">Log out</A>

Example (logout):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0logout-response.dtd)
An ordinary HTTP connection.	GET /v1.0/general/id/logout;
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

7.9.2 Response format (logout):

Explanation **-----------**	**Format of information sent** -------------------------- (http://select/v1.0/logout.dtd)
The evaluation are returned, one rating service at a time.	<!ELEMENT logout-response EMPTY>
Whether all the rating labels were accepted, or some of them, or none of them.	<!ATTLIST logout-response accepted (true | false) 'true'
You tried to logout, but you were not logged in.	not-logged-in (true | false) 'false'
End of XML attribute list	>

Example (logout response):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/logout-response.xml)
HTTP header	HTTP/1.1 200 OK Date: Sun, 25 Jul 1999 13:32:18 +0200 Server: Apache/1.2.4 Last-Modified: Sun, 25 Jul 1999 13:32:18 +0200 ETag: "437e5-98-3531f2e3" Content-Length: 152 Accept-Ranges: bytes Connection: close Content-Type: application/xml
Max-age="0" resets the cookie.	Set-cookie: session="1234567890123456";Domain="select";Path="/v1.0/general/id/";Max-age="0"
A blank line to mark the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE logout-response SYSTEM "http://select/v1.0/logout-response.dtd">
Start and end of login-response for a rejected login.	<logout-response accepted="false"/>

7.10 Get-Atomic-Ratings

Summary: The *get-atomic-ratings* operation retrieves atomic ratings done by one or more named raters on one or more resources. It can be used by a user agent to find out if this user has already rated this resource. It might also be used in peer rating, where person A wants to find items rated highly by named individuals B and C.

Access control: The ratings made by a certain user can only be seen by that user, i.e. after logging in as that user. A person may however, in his/her personal profile, specify that other people can see his/her ratings. Get-atomic-ratings on a list of people may only be done in the following cases (i) all the people have specified in their profile that their ratings may be seen by other people, or (ii) the requestor is a certified filtering agent which will not divulge the personal ratings to a person, or (iii) the list of users is larger than ten, in this case, the atomic ratings are returned without identification of who made which rating.

Input data: A URI for the rated resource, and a list of one or more people, whose atomic ratings on this resource are wanted.

Output data: A list of atomic ratings, with or without identification of who made them, or an error code.

Base protocol: XML.

7.10.1 Query format (get-atomic-ratings):

The get-atomic-ratings query is an HTTP POST operation, whose body is an XML resource containing the query, sent to http://select/v1.0/general/id/evaluator.

Note: You must be logged in, to perform this operation, even if you only are going to retrieve anonymous ratings.

Explanation **-----------**	**Information sent** **----------------**
Connect to the SELECT server	POST /v1.0/general/id/get-ratings HTTP/1.1
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
The format of the query is XML.	Content-Type: Application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

The body of the query is an XML [XML1], [XML2] resource. The XML Resource Type Declaration for this XML resource is:

Explanation **-----------**	**Format of information sent** -------------------------- (http://select/v1.0/get-atomic-ratings.dtd)
	<!ELEMENT get-atomic-ratings (location+, rater+, labelname*)>
Start of attribute list for get-atomic-ratings.	<!ATTLIST get-atomic-ratings
Restrict the retrieval to only ratings done in a certain context.	context (general | business | leisure | shopping | research | politics | all) 'all'
Whether only ratings made by this rater (identified or pseudonymous) can be retrieved. Note: If you set this setting to "all" then you will get back ratings without identity or date on them.	whose-ratings (own | all) 'own'
End of the list of XML attributes.	>
Each URI to be evaluated is a free text field containing the URI of the resource to be evaluated.	<!ELEMENT location EMPTY>
Start of attribute list for rater.	<!ATTLIST location
Raterid or pseudonym. If Raterid is given, only ratings made non-anonymously for this user are returned, if pseudonym is given, only ratings made under this pseudonym are returned. Thus, raterid and pseudonym are treated as two different raters. One exception: A rater has access rights to retrieve own ratings made both anonymously and non-anonymously, but the rater must then list both raters in two "rater" elements in the request. Note: Possibly, processes with special privileges may be allowed to retrieve ratings made by different people and anonymous ratings?	uri CDATA #REQUIRED
End of the list of XML attributes.	>
Identify whose ratings are requested.	<!ELEMENT rater EMPTY>

Start of attribute list for rater. Omitted if you want all ratings, made by anyone, in un-identified format.	<!ATTLIST rater
Raterid or pseudonym or the fixed string "anonymous" to retrieve anonymous ratings. If Raterid is given, only ratings made non-anonymously for this user are returned, if pseudonym is given, only ratings made under this pseudonym are returned. Thus, raterid and pseudonym are treated as two different raters. One exception: A rater has access rights to retrieve own ratings made both anonymously and non-anonymously, but the rater must then list both raters in two "rater" elements in the request.	raterid CDATA #REQUIRED
End of the list of XML attributes.	>
List of requested rating descriptors. If no list is specified, this means that all available ratings are requested.	<!ELEMENT labelname EMPTY>
Start of attribute list for rater.	<!ATTLIST labelname
Raterid or pseudonym. If Raterid is given, only ratings made non-anonymously for this user are returned, if pseudonym is given, only ratings made under this pseudonym are returned. Thus, raterid and pseudonym are treated as two different raters. One exception: A rater has access rights to retrieve own ratings made both anonymously and non-anonymously, but the rater must then list both raters in two "rater" elements in the request. To retrieve ratings made by other people in de-identified format, enter the name as the string "other".	name CDATA #REQUIRED
End of the list of XML attributes.	>

Example of a body (get-atomic-ratings):

Explanation -----------	Information sent ---------------- (http://select/v1.0/get-atomic-ratings.dtd)
Start.	<?xml version="1.0"?> <!DOCTYPE get-atomic-ratings SYSTEM "http://select/v1.0/get-atomic-ratings.dtd"> <get-atomic-ratings context="leisure">
List of locations, for which ratings are retrieved.	<location uri="http://www.body.com/toes"/> <location uri="http://www.face.com/eyes"/>
Raters, whose ratings are requested.	<rater raterid="jpalme@dsv.su.se"/> <rater raterid="father.christmas@northpole.com"/>
Which rating labels are requested.	<labelname name="select-reader-interest-rating"/> <labelname name="keywords"/>
End of get-atomic-ratings.	</get-atomic-ratings>

7.10.2 Response format (get-atomic-ratings-response):

The response is an XML [XML1], [XML2] document. The XML Resource Type Declaration for this XML resource is:

Explanation -----------	Format of information sent -------------------------- (http://select/v1.0get-atomic-ratings-response.dtd)
Reference to data structure defined in a separate DTD file. Further information, see section 7.2.	<!ENTITY % atomic-rating SYSTEM "http://select/v1.0/atomic-rating.dtd">

Import DTD from separate DTD file atomic-rating.dtd.	%atomic-rating;
The evaluation are returned, one rating service at a time.	<!ELEMENT get-atomic-ratings-response (rejection+ | atomic-rating+)>
If only some of the settings were accepted, here is a list of those not accepted. The #PCDATA can contain a human-readable description of the refusal reason in the preferred language of the user doing the registration (not always the language of the user being registered).	<!ELEMENT rejection (#PCDATA)>
XML attributes for refuse-reason.	<!ATTLIST rejection
Why the attribute was rejected.	refuse-reason (
	authorisation | bad-syntax | no-such-attribute | no-ratings-available | not-logged-in | other-reason
End of refuse-reason.	) #REQUIRED
Refused value of this attribute.	refused-value CDATA #IMPLIED
End of XML attribute list.	>

Example 1 (get-atomic-ratings-response):

Note: This response is sent in the case where the ISCN server had no ratings for any of the resources requested, so that only ratings from the select general ratings server are returned.

Explanation **-----------**	**Information sent** **----------------**
HTTP response header	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE get-atomic-ratings-response SYSTEM "http://select/v1.0/get-atomic-ratings-response.dtd">
Start of get-atomic-ratings-response for one resource.	<get-atomic-ratings-response>
First rating returned.	<atomic-rating raterid-or-pseudonym="jpalme@dsv.su.se" rating-engine="select/select-proxy-1" location="http://www.body.com/eyes" rating-date="31 Jul 1999" rater-competence="user" rater-type="manual" rater-trust="registered" message-id="990815113350*jpalme@dsv.su.se">
First rating descriptor.	<rating-value type="select-reader-interest-rating" value="good"/>
Second rating descriptor.	<rating-value type="adult" value="false"/>

Third rating descriptor.	<rating-value
	  type="context"
	  value="leisure"/>

End of data.	</atomic-rating>

Second rating returned.	<atomic-rating
	
	raterid-or-pseudonym="father.christmas@northpole.com"
	  rating-engine="select/select-proxy-1"
	  location="http://www.body.com/eyes"
	  rating-date="17 Aug 1999"
	  rater-competence="expert"
	  rater-type="manual"
	  rater-trust="registered"
	  message-id="990815113350*jpalme@dsv.su.se">

First rating descriptor.	<rating-value
	  type="select-reader-interest-rating"
	  value="87"/>

Second rating descriptor.	<rating-value
	  type="adult"
	  value="false"/>

Third rating descriptor.	<rating-value
	  type="context"
	  value="leisure"/>

End of data	</atomic-rating>
	</get-atomic-ratings-response>

Example 2 (get-atomic-ratings-response rejection):

Explanation -----------	Information sent ----------------
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE get-atomic-ratings-response SYSTEM "http://select/v1.0/get-atomic-ratings-response.dtd">
Start of server list.	<get-atomic-ratings-response>
Start of ratings for one resource to be rated.	<rejection refuse-reason="not-logged-in"/>
End of evaluate-response report and end of file.	</get-atomic-ratings-response>

7.11 Simple-Search Operation

Issue: Should this really be in the standard? Is this not a user interface issue, since it is specified as an HTML search form below?

Summary: Find web pages satisfying a query and which are highly rated.

Access control: No access control for basic rating. Rating based on a particular users interest and values may be available only if preceded by a login operation for this particular user.

Input data: The user specifies the query by filling in a query form. Simple search, when the personalised checkbox is unchecked, is always made on the general-rating derived descriptor. When the Personalized search checkbox is checked, the general-rating is made using a default personal-rating derived descriptor, which actually returns different values for each user. If the user is unknown, Personalized search will return an error message.

Output data: A HTML page or an XML document with a list of found pages sorted according to rating and relevance.

Base protocol: HTML application/x-www-form-urlencoded for the request, and HTML or XML for the response.

7.11.1 Query format (simple-search-query):

The simple-search query is an HTTP GET operation with the query after "?" in the URI.

The query is the same as would be sent with the following HTML form:


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<TITLE>SELECT Search Query</TITLE>
<style type="text/css">
<!--
p {  font-family: Verdana, Arial, Helvetica, Geneva, sans-serif;
font-size: 10pt}
td {  font-family: Verdana, Arial, Helvetica, Geneva, sans-serif;
font-size: 10pt}
-->
</style></HEAD>
<BODY bgcolor="#FFFFFF">
<FORM ACTION="http://www.dsv.su.se/~jpalme/test/echo.cgi" METHOD=get
NAME="searchform">
  <table border="0" cellspacing="0" cellpadding="2" align="center">
    <tr bgcolor="#6633CC" align="center">
      <td rowspan=5 valign="top" width="121" align="center">
        <div align="left"><font color='white'> Search: <br>
          <input type="checkbox" name="search"
                 value="internet" checked>
          Internet <br>
          <input type="checkbox" name="search" value="select"
                 checked>
          Select directory <br>
          <input type="checkbox" name="search" value="news"
                 checked>
          News </font></div>
        <font color='white'>
```

```
        <hr width="70" align="left">
        <div align="left">
          <input type="checkbox" name="unseen" value="yes">
          Only unseen</div>
        </font></td>
      <td colspan=5 rowspan="2"><font color='white'>
            Search query:
        <INPUT SIZE=54 MAXLENGTH=256 NAME="query" value="">
         
        <input type="submit" name="Search" value="Search">
        </font></td>
    </tr>
    <tr bgcolor="#FFFFCC">
      <td width="21"><font color="#FFFFFF"></font></td>
    </tr>
    <tr bgcolor="#CCFF99">
      <td valign="top" width="163" align="center" > Limit to
Country:<br>
        <input type="text" name="textfield">
      </td>
      <td valign="top" width="122" > <b>Limit to Language:</b><br>
        <select name="lang" size=1>
          <option value="world" selected>Any
          <option value="welsh">Cymraeg
          <option value="dansk">Dansk
          <option value="deutsch">Deutsch
          <option value="english">English
          <option value="espa&ntilde;ol">Espa&ntilde;ol
          <option value="fran&ccedil;ais">Fran&ccedil;ais
          <option value="italiano">Italiano
          <option value="magyar">Magyar
          <option value="nederlands">Nederlands
          <option value="norsk">Norsk
          <option value="portugu&ecirc;s">Portugu&ecirc;s
          <option value="suomi">Suomi
          <option value="svenska">Svenska
        </select>
      </td>
      <td valign="top" width="109" >
        <p align="center"> <b>Result format:</b><br>
          <input type="radio" name="resultformat"
                 value="html" checked>
          HTML
          <input type="radio" name="resultformat" value="xml">
          XML </p>
      </td>
      <td valign="top" width="24">
```

```
        <div align="right">
          <input type="checkbox" name="personalized" value="yes">
        </div>
      </td>
      <td valign="top" width="145" >
        <p>Peer search</p>
        <p><b>Max no of docs:</b>
          <input type="text" name="maxno" size="4"
                 maxlength="20" value="50">
        </p>
      </td>
      <td width="21" bgcolor="#FFFFFF"> </td>
    </tr>
    <tr bgcolor="#FFFFCC">
      <td valign="middle" colspan="5" align="center"> Context:
        <input type="checkbox" name="context" value="yes" checked>
        general
        <input type="checkbox" name="business" value="yes" checked>
        business
        <input type="checkbox" name="leisure" value="yes" checked>
        leisure
        <input type="checkbox" name="shopping" value="yes" checked>
        shopping
        <input type="checkbox" name="research" value="yes" checked>
        research
        <input type="checkbox" name="politics" value="yes" checked>
        politics<br>
      </td>
      <td width="21" rowspan="2"><font color="#FFFFFF"></font></td>
    </tr>
    <tr bgcolor="#FFFFCC">
      <td valign="middle" colspan="5" align="center"
          bgcolor="#6633CC"> <font color="#FFFFFF">
        <input type="checkbox" name="Use my keywords"
               value="Keyworduse" checked>
        Use my interest profile  
        <input type="checkbox" name="usekeywords"
               value="usemykeywords" checked>
         Use my keywords  
        <input type="checkbox" name="onlymanual"
               value="onlymanual">
         Use only manual keywords and profile</font></td>
    </tr>
  </table>
</FORM>
</BODY></HTML>
```

If the user check to "Use my interest profile" or "Use my keywords", then that user can, but need not fill in any "Search query". If the user does not fill in any "Search Query" but checks "Only unseen" and "Use my interest profile" or "Use my keywords", then this will be a search for highly-rated new, by this user unseen information. Note that by checking "News", a search for news articles is done and the result may be presented on the web, even though the rating of these web articles was done through a newsreader and not through a web interface.

By "Peer search" is meant search, where higher value is given to ratings provided by people with similar interests and values as yourself.

Example of query string:

(filter OR "SELECT rating") AND EU&domain=world&language=world

which with URI encoding will become:

search=internet&search=select&search=news&query=%28filter+OR+%22SELECT+rating%22%29+AND+EU&Search=Search&textfield=&lang=world&resultformat=html&context=yes&business=yes&leisure=yes&shopping=yes&research=yes&politics=yes

Example of a simple-search query

Query is sent to the following URL for the SELECT general service:

http://select/v1.0/general/simple-search?query=

Explanation -----------	**Information sent** ----------------
Connect to the SELECT server "format" can be either "xml" or "html" and specifies in which format the response is to be delivered	GET /v1.0/general/simple-search?search=internet&search=select&search=news&query=%28filter+OR+%22SELECT+rating%22%29+AND+EU&Search=Search&textfield=&lang=world&resultformat=html&context=yes&business=yes&leisure=yes&shopping=yes&research=yes&politics=yes HTTP/1.1
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml

This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

7.11.2 Response format (simple-search-response):

The simple-search response can be in either XML or HTML format depending on the request. If no format was specified in the request, HTML is the default format. The response contains a list of resources matching the query and sorted by rating-value. This standard only specifies the XML response format, the HTML response format is not standardized.

The XML Resource Type Declaration for this XML resource is:

Explanation **-----------**	**Format of information sent** -------------------------- (http://select/v1.0/simple-search-response.dtd)
The evaluation are returned, one rating service at a time.	<!ELEMENT simple-search-response (error | resource+)>
	<!ELEMENT error (#PCDATA)>
	<!ATTLIST error
If rating was rejected, explanation why. See Refusal reasons, chapter 7.5.2.1.	refuse-reason (bad-syntax | access-control | other) 'access-control'
End of XML attribute list.	>
If only some of the rating values were rejected, this element is used to list the rejected rating values. The #PCDATA contains the summary or keywords or some other description of the found resource.	<!ELEMENT resource (#PCDATA)>
	<!ATTLIST resource
Some kind of computed rating value.	rating CDATA #REQUIRED
	title CDATA #IMPLIED
URI of the found resource.	uri CDATA #REQUIRED
End of XML attribute list.	>

Example 1 (positive simple-search response):

Explanation **-----------**	**Information sent** **----------------**
HTTP response header	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE simple-search-response SYSTEM "http://select/v1.0/simple-search-response.dtd">
	<simple-search-response>
	<resource rating="88" title="Kenyan flowers" uri="http://www.flowers.com/kenya/">
	An overview of flowers found in Kenya.
	</resource>
	<resource rating="78" title="Kiwi flowers" uri="http://www.flowers.com/kiwi/">
	An overview of flowers found in Kiwi.
	</resource>
	</simple-search-response>

Example 2 (negative simple-search response):

Explanation **-----------**	**Information sent** **----------------**
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE simple-search-response SYSTEM "http://select/v1.0/simple-search-response.dtd">
All ratings were not accepted.	<simple-search-response>
Rating-label rejected, this server does not accept ratings in the leisure context.	<error>You are not allowed to make this search.</error>
End of simple-search-response.	</simple-search-response>

7.12 Advanced-Search Operation *(Not yet ready)*

Summary: Find web pages satisfying a query and which are highly rated.

Access control: No access control for basic rating. Rating based on a particular users interest and values may be available only if preceded by a login operation for this particular user.

Input data: Some general-purpose search format, based on SQL or some other search language. The advanced search should especially allow the needs of other modules.

Required functionality:

1. It should be possible to search on all derived and atomic ratings. Example of use: The NLP modules need a way of getting a list of which documents are to be rated by the NLP modules. Can this be done through a

variant of the advanced-search operation?

2. It should be possible to retrieve all ratings on resources with a particular author, including ratings with a particular author sent to a particular newsgroup.

Output data: A HTML page or an XML document with a list of found pages sorted according to rating and relevance.

Base protocol: HTML application/x-www-form-urlencoded for the request, and HTML or XML for the response.

7.12.1 Query format (advanced-search-query):

The advanced-search query is an HTTP POST operation, whose body is an XML resource containing the profile, sent to the profiles cgi-script in the server for this particular rating service. Example: "http://select/v1.0/general/id/search".

Explanation -----------	**Information sent** ----------------
Connect to the SELECT server	POST /v1.0/general/id/search HTTP/1.1
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
The format of the query is XML.	Content-Type: Application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

The body of the operation is an XML [XML1], [XML2] resource. The XML Resource Type Declaration for this XML resource is:

Explanation -----------	**Format of information sent** -------------------------- (http://select/v1.0/advanced-search.dtd)
Not yet ready	

Example of a advanced-search query

Explanation **-----------**	**Format of information sent** -------------------------- (http://select/v1.0/advanced-search.dtd)
HTTP header.	POST /v1.0/general/id/advanced-search HTTP/1.1 Host: select Accept: application/xml Content-Type: Application/xml Cookie: session="012345678901234354"
A blank line to mark the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE advanced-search SYSTEM "http://select/v1.0/advanced-search.dtd">
Not yet ready	

7.12.2 Response format (advanced-search-response):

The response format for the advanced-search is the same as the response format for the simple search, described in section 0.

7.13 Evaluate Operation

Summary: Get the ratings for a list of URIs.

Access control: No access control for basic rating. Rating based on a particular user's interest and values may be available only if preceded by a login operation for this particular user.

Input data: A list of URIs and a list of services. For each service, a list of aggregate rating labels are listed. Note that only aggregate ratings, not atomic ratings, can be found with this operation. If N URIs, M services and V label types are listed, then NxMxV rating labels are returned.

Output data: A list of rating labels.

Base protocol: HTTP and XML.

Issue: ***Is a "streaming" version of this operation needed? By streaming is meant a version in which the URIs to process are sent to the server in parallel with the server returning responses, so that responses for the first URIs are returned before the last URIs have been sent to the server for evaluation.***

7.13.1 Query format (evaluate-query):

The evaluate query is an HTTP POST operation, whose body is an XML resource containing the query, sent to http://select/v1.0/general/evaluator

Explanation -----------	Information sent ----------------
Connect to the SELECT server.	POST /v1.0/general/evaluator HTTP/1.1
To the HTTP server "select" port 80.	Host: select
Only files in the format application/xml are accepted.	Accept: application/xml
The format of the query is XML.	Content-Type: Application/xml
This user has connected to this server before, and a cookie identifies the session.	Cookie: session="1234567890123456"

The body of the query is an XML [XML1], [XML2] resource. The XML Resource Type Declaration for this XML resource is:

<table>
<tr><th>Explanation
-----------</th><th>Format of information sent

(http://select/v1.0/evaluate-query.dtd)</th></tr>
<tr><td>A list of locations to be evaluated, followed by a list of services to evaluate these locations. The returned response will be L x S rating labels, if L is the number of locations and S the number of services.</td><td><!ELEMENT evaluate-query (location+, service+)></td></tr>
<tr><td>Start of attribute list for evaluate-query.</td><td><!ATTLIST evaluate-query</td></tr>
<tr><td>Whether rating are to be personalised by comparison to other people with similar views to myself.</td><td>personal (true | false)'false'</td></tr>
<tr><td>True means that the responses are sorted in rating priority order. False means that the responses are returned in the order they were given in the request.</td><td>sort (true | false)'true'</td></tr>
<tr><td>Restrict the evaluation to only ratings done in a certain context.</td><td>context (general | business | leisure shopping | research | politics | all) 'all'</td></tr>
<tr><td>End of the list of XML attributes.</td><td>></td></tr>
<tr><td>Each URI to be evaluated is a free text field containing the URI of the resource to be evaluated.</td><td><!ELEMENT location (#PCDATA)></td></tr>
<tr><td>Each service description is a free text field containing the URI of the service.</td><td><!ELEMENT service (label* | collection-name)></td></tr>
<tr><td>Start of attribute list for service.</td><td><!ATTLIST service</td></tr>
<tr><td>Identification of the service by its URI.</td><td>location CDATA #REQUIRED</td></tr>
</table>

End of the list of XML attributes.	>
List of requested descriptors. If no list is specified, this means that all available descriptors are requested. Only aggregate ratings can be requested, not atomic ratings.	<!ELEMENT label (#PCDATA)>
Start of attribute list for label.	<!ATTLIST label
If match is true, then all labels whose name begin with the given string are retrieved. For example, with match=true and the label value "keywords", labels of derived descriptors like "keywords-tropical" and "keywords-flowers" might be retrieved.	match (false | true) 'false'
End of the list of XML attributes.	>
Instead of listing the labels to be retrieved, it is possible to just specify the name of a collection, to retrieve the labels specified in this collection.. The collection must be a collection specified in the service-description of the service used.	<!ELEMENT collection-name EMPTY> <!ATTLIST collection-name name CDATA #REQUIRED >

Example of a body (evaluate-query):

Explanation **-----------**	**Information sent** ---------------- (http://select/v1.0/evaluate-query.xml)
Start.	<?xml version="1.0"?> <!DOCTYPE evaluate-query SYSTEM "http://select/v1.0/evaluate-query.dtd"> <evaluate-query>
List of locations to be evaluated.	<location>http://www.body.com/toes</location> <location>http://www.face.com/eyes</location>
List of services whose evaluations are requested. For each service, the descriptors requested are listed. For the general service, reader-quality and reader-interest-ratings are requested, for the iscn service, all available descriptors are requested.	<service location="http://select/v1.0/general/general-service-description.xml"> <label>select-reader-quality-rating</label> <label>select-reader-interest-rating</label> <label match="true">keywords</label> </service> <service location="http://select/v1.0/general/iscn-service-description.xml" /> <service location="http://select/v1.0/general/flower-lovers-service-description.xml"> <collection-name name="instant-ratings"/> </service>
End of evaluate-query.	</evaluate-query>

7.13.2 Response format (evaluate-response):

The response is an XML [XML1], [XML2] document. The XML Resource Type Declaration for this XML resource is:

Explanation **-----------**	**Format of information sent** **--------------------------** (http://select/v1.0/evaluate-response.dtd)
The evaluation are returned, one rating service at a time.	<!ELEMENT evaluate-response (rejection | resource+)>
Start of list of attributes for the evaluate-service element.	<!ATTLIST evaluate-response
URI of the service, only used if all ratings returned are from the same service.	service CDATA #IMPLIED
Whether rating are to be personalised by comparison to other people with similar views to myself.	personal (true | false) 'false'
True means that the responses are sorted in rating priority order. False means that the responses are returned in the order they were given in the request.	sort (true | false) 'true'
End of the list of XML attributes.	>

	<!ELEMENT rejection EMPTY>
	<!ATTLIST rejection
Reject reason, no default value.	reject-reason (not-logged-in | bad-syntax | not-found | authorisation-failure | other-reason) #IMPLIED
End of XML attributes for "rejection".	>
	<!ELEMENT resource (label*)>
URI of the rated resource.	<!ATTLIST resource location CDATA #REQUIRED
End of XML attributes for "resource".	>
Start of a ratings label. Note: If no label is available, then no labels are specified.	<!ELEMENT label EMPTY>
Start of attribute list for label.	<!ATTLIST label
URI of the service providing this label. This attribute may be omitted in the following two cases: (a) if all ratings come from the same service, and this service was specified as an attribute to the evaluate-response. (b) in a series of labels from the same service on the same resource, only the first need specify the service.	service CDATA #IMPLIED

Default descriptor format is numerical. Alternative descriptors are words (list of keywords etc.) or text (any plain UTS-8 text) or date (in mail header format, for example "29 Jul 1999".	format (numerical | words | text | date) 'numerical'
Name of a descriptor, either its transmit-as or short-form name, as specified in the rating service description for the rating service used.	name CDATA #REQUIRED
The format of the value depends on the type, as specified in the rating service description.	value CDATA #REQUIRED
The confidence (number of evaluators) behind this value.	confidence CDATA #IMPLIED
This rating value is only valid in a certain context.	context (general | business | leisure | shopping | research | politics | all)'all'
End of attribute list.	>

Example 1 (evaluate response):

Note: This response is sent in the case where the ISCN server had no ratings for any of the resources requested, so that only ratings from the select general ratings server are returned.

Explanation **-----------**	**Information sent** **----------------**
HTTP response header	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE evaluate-response SYSTEM "http://select/v1.0/evaluate-response.dtd">
Start of evaluate-response for one resource.	<evaluate-response>
Start of ratings for one resource to be rated.	<resource location="http://www.body.com/toes">
One rating descriptor value for this resource.	<label
	service="http://select/v1.0/general/general-service-description.xml"
Confidence of this value.	confidence="12"
Type of label value.	format="numerical"
Name of this label (either transmit-as or short-name).	name="select-reader-quality-rating"
Value of this descriptor.	value="88"
Restricted context of this rating value.	context="leisure"
End of this label.	/>

Another rating descriptor value.	<label
Confidence of this label.	confidence="57"
Type of label value.	format="numerical"
A derived attribute containing a frequency count.	name="keywords-tropical"
Number of people who have assigned the keyword "tropical" to this resource.	value="3"
Restricted context of this rating value.	context="leisure"
End of this label.	/>
Another rating descriptor value.	<label
Confidence of this label.	confidence="33"
Type of label value.	format="numerical"
Name of this label (either transmit-as or short-name).	name="select-reader-interest-rating"
Value of this label.	value="78"
Restricted context of this rating value.	context="leisure"
End of this label.	/>
One rating descriptor value for this resource.	<label
	service="http://select/v1.0/general/iscn-service-description.xml"
Type of label value.	format="numerical"
Confidence of this value.	confidence="12"

Name of this label (either transmit-as or short-name).	name="scientific-relevance"
Value of this descriptor.	value="88"
Restricted context of this rating value.	context="research"
End of this label.	/>
End of list of all labels for this resource.	</resource>
Start of ratings for one resource to be rated.	<resource location="http://www.body.com/toes">
One rating descriptor value.	<label
	service="http://select/v1.0/general/general-service-description.xml"
Confidence of this label.	confidence="55"
Type of label value.	format="numerical"
Name of this label (either transmit-as or short-name)	name="select-reader-quality-rating"
Value of this label.	value="88"
End of this label.	/>
One rating descriptor value.	<label
Confidence of this label.	confidence="33"
Default descriptor format is numerical. Alternative descriptors are words (list of keywords etc.) or text (any plain UTS-8 text) or date (in mail header format, for example "29 Jul 1999".	format="numerical"

Name of this label (either transmit-as or short-name).	name="select-reader-interest-rating"
Value of this label.	value="78"
End of this label.	/>
End of list of all labels for this resource.	</resource>
End of evaluate-response report and end of file.	</evaluate-response>

Example 2 (evaluate response rejection):

Explanation **-----------**	**Information sent** **----------------**
HTTP response header.	HTTP/1.1 200 OK Content-Length: 569 Content-Type: application/xml Server: Select 1.0 Date: 7 July 1999 19:58:23 +0200
A blank line to indicate the end of the HTTP header.	
Identifies that this is in XML format.	<?xml version="1.0"?>
References the Resource Type Declaration (DTD) file specifying the syntax for this XML resource.	<!DOCTYPE evaluate-response SYSTEM "http://select/v1.0/evaluate-response.dtd" >
Start of server list.	<evaluate-response>
Start of ratings for one resource to be rated.	<rejection reject-reason="not-logged-in"/>
End of evaluate-response report and end of file.	</evaluate-response>

7.14 Exchange-Ratings-Data *(not yet ready)*

Summary: This operation is used between two select servers, in order to replicate information in their data bases.

Issues:

Access control:

Input data:

Output data:

Base protocol:

7.14.1 Query format (exchange-ratings-data):

Example (replicate-ratings):

Explanation **-----------**	**Information sent** **----------------**
Not ready	

7.14.2 Response format (exchange-ratings-data):

Explanation **-----------**	**Format of information sent** -------------------------- (http://select/v1.0/exchange-ratings-data.dtd)
Not ready	

Example (replicate-ratings response):

Explanation **-----------**	**Information sent** **----------------**
Not ready	

8. The SELECT general service description

A SELECT general service description file contains

- A list of services
- Per service
- Admistrative information about the service accessible on the server
- Name
- Maintainer
- Website about the service
- Textual description in natural language (possible in multiple languages)
- A list of categories
- Per category
- A textual description of the category (possible in multiple languages)
- A name for the category
- Rater type (human or computer generated rating)
- The datatype of the ratings for this category (a label, keyword, value or derived category)
- Depending on the datatype
- Value:
  How the category should be displayed on screen ("none" if not possible)
  The calculationmethod used to calculate the instant rating value of a resource for this category
  A minimum and maximum value for the rating values in this category
- Label :
  How the category should be displayed on screen ("none" if not possible)
  The calculationmethod used to calculate the instant rating value of a resource for this category
  A list of labels for the category. Per Label
- A value that corresponds to the description contained in the textual or iconic labels.
- A list of Textual and/or Iconic labels (possible in multiple languages)
- Derived:
  The calculationmethod used to calculate the instant rating value of a resource for this category
  A number of categories from which the value of an instant rating of this category is derived.
  A list of labels that describe how to map the value of an instant rating of this category back to natural language.
  Per label
- A minumum and maximum value. If the rating falls between these 2 values, the associated textual label is selected.
- Keyword:
  The calculationmethod used to calculate the instant rating

value of a resource for this category
- A list of imported categories : categories of other services that are imported into this service
- The classname of the Java class that starts the agents associated with the service.

All this translates to the XML document type definition looks like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!ELEMENT services (service*)>
<!ELEMENT description EMPTY>
<!ATTLIST description
   language CDATA #REQUIRED
   text     CDATA #REQUIRED
>
<!ELEMENT import EMPTY>
<!ATTLIST import
   service  CDATA #REQUIRED
   category CDATA #REQUIRED
>
<!ELEMENT service (description+, category+, import*, agentinit?)>
<!ATTLIST service
   id          CDATA #REQUIRED
   URI         CDATA #REQUIRED
   server      CDATA #REQUIRED
   maintainer  CDATA #REQUIRED
   anonymous   (allowed | forbidden) 'allowed'
>
<!ELEMENT agentinit EMPTY>
<!ATTLIST agentinit
  classname CDATA #REQUIRED
>
<!ELEMENT category (description+, (labelcategory | valuecategory |
keywordcategory | derivedcategory))>
<!ATTLIST category
   id         CDATA #REQUIRED
   rater-type (human | computer) 'computer'
   type       (label | value | keyword | derived) 'label'
>
<!ELEMENT labelcategory (label+)>
<!ATTLIST labelcategory
   gui         (buttons | radiobuttons | list | icons | none) 'none'
   calculation CDATA #REQUIRED
>
<!ELEMENT label (description*, icon*)>
<!ATTLIST label
```

```
   value  CDATA #REQUIRED
>
<!ELEMENT icon EMPTY>
<!ATTLIST icon
   language CDATA #REQUIRED
   text     CDATA #REQUIRED
>
<!ELEMENT valuecategory EMPTY>
<!ATTLIST valuecategory
   gui         (slider | buttons | none) 'none'
   min         CDATA #IMPLIED
   max         CDATA #IMPLIED
   upperlimit  (yes | no) 'no'
   calculation CDATA #REQUIRED
>
<!ELEMENT keywordcategory EMPTY>
<!ATTLIST keywordcategory
   calculation CDATA #REQUIRED
>
<!ELEMENT derivedcategory (derivedfrom+, derivedlabel*)>
<!ATTLIST derivedcategory
   calculation CDATA #REQUIRED
>
<!ELEMENT derivedfrom EMPTY>
<!ATTLIST derivedfrom
   service    CDATA #REQUIRED
   category   CDATA #REQUIRED
>
<!ELEMENT derivedlabel (description*)>
<!ATTLIST label
   min  CDATA #REQUIRED
   max  CDATA #REQUIRED
>
```

8.1 Example

An example of all this is the service description file of the SELECT test server:

```
<?xml version="1.0"?>
<!DOCTYPE services SYSTEM "services.dtd">
<services>
  <!-- The SELECT test service -->
  <!-- ********************** -->
  <service
     id         = "test"
```

```
    URI        = "http://samson.aszi.sztaki.hu/SELECT"
    server     = "samson.aszi.sztaki.hu"
    maintainer = "micsik@sztaki.hu"
    anonymous  = "allowed">
    <description language = "en" text = "SELECT test service">
    </description>
    <description language = "nl" text = "SELECT test dienst">
    </description>
    <!-- Quality of the document according to the user -->
    <category
       id         = "contents"
       rater-type = "human"
       type       = "label">
       <description language = "en"
       text = "Quality of the content of the document">
      </description>
       <description language = "nl"
       text = "Kwaliteit van de inhoud van het document">
      </description>
       <labelcategory
          gui         = "buttons"
          calculation = "average">
          <label value = "1">
             <description language = "en"
             text = "Awful"></description>
             <description language = "nl" text = "Verschikkelijk">
             </description>
             <icon        language = "en" text =
             "http://samson.aszi.s ztaki.hu/SELECT/icons/1star.gif">
            </icon>
          </label>
          <label value = "2">
             <description language = "en"
             text = "Mediocre"></description>
             <description language = "nl"
             text = "Middelmatig">
             </description>
             <icon        language = "en" text =
             "http://samson.aszi.sztaki.hu/SELECT/icons/2star.gif">
            </icon>
          </label>
          <label value = "3">
             <description language = "en" text = "OK"></description>
             <description language = "nl" text = "OK"></description>
             <icon        language = "en" text =
             "http://samson.aszi.sztaki.hu/SELECT/icons/3star.gif">
            </icon>
```

```
            </label>
            <label value = "4">
               <description language = "en" text = "Good"></description>
               <description language = "nl" text = "Goed"></description>
               <icon        language = "en" text =
               "http://samson.aszi.sztaki.hu/SELECT/icons/4star.gif">
              </icon>
            </label>
            <label value = "5">
               <description language = "en" text = "Great">
              </description>
               <description language = "nl" text = "Geweldig">
              </description>
               <icon        language = "en" text =
              "http://samson.aszi.sztaki.hu/SELECT/icons/5star.gif">
              </icon>
            </label>
         </labelcategory>
      </category>
      <!-- Style of the document according to NLP -->
      <category
         id          = "style"
         rater-type = "computer"
         type        = "value">
         <description language = "en" text =
         "The quality of the document according to the NLP modules">
         </description>
         <description language = "nl" text =
         "De kwaliteit van het document volgen de NLP modules">
         </description>
         <valuecategory
             gui         = "none"
             min         = "0"
             max         = "10"
             upperlimit  = "yes"
             calculation = "NLP"/>
      </category>
      <!-- Time spent reading the document -->
      <category
         id          = "readtime"
         rater-type = "computer"
         type        = "value">
         <description language = "en" text =
         "The time spent reading this document in seconds">
         </description>
         <description language = "nl" text =
         "De tijd waarin het document gelezen werd in seconden">
```

```
        </description>
        <valuecategory
           gui         = "none"
           min         = "0"
           max         = "600"
           upperlimit  = "no"
           calculation = "median"/>
     </category>
     <!-- Human added keywords -->
     <category
        id         = "keywords"
        rater-type = "human"
        type       = "keyword">
        <description language = "en"
        text = "Human added keywords"></description>
        <description language = "nl"
        text = "Kernwoorden die door de gebruiker werden toegevoegd">
        </description>
        <keywordcategory
           calculation = "count"/>
     </category>
     <!-- Overall quality of the document -->
     <category
        id         = "quality"
        rater-type = "computer"
        type       = "derived">
        <description language = "en"
        text = "Overall quality of the document"></description>
        <description language = "nl" text =
       "Algemene kwaliteit van het document"></description>
        <derivedcategory
           calculation = "statistics">
           <derivedfrom service = "test" category = "contents"/>
           <derivedfrom service = "test" category = "style"/>
           <derivedlabel min = "0" max = "1">
              <description language = "en"
              text = "Awful"></description>
              <description language = "nl"
              text = "Verschrikkelijk"></description>
           </derivedlabel>
           <derivedlabel min = "1" max = "2">
              <description language = "en"
              text = "Mediocre"></description>
              <description language = "nl"
              text = "Middelmatig"></description>
           </derivedlabel>
           <derivedlabel min = "2" max = "3">
```

```
            <description language = "en" text = "OK"></description>
            <description language = "nl" text = "OK"></description>
         </derivedlabel>
         <derivedlabel min = "3" max = "4">
            <description language = "en" text = "Good"></description>
            <description language = "nl" text = "Goed"></description>
         </derivedlabel>
         <derivedlabel min = "4" max = "5">
            <description language = "en"
            text = "Great"></description>
            <description language = "nl"
            text = "Geweldig"></description>
         </derivedlabel>
      </derivedcategory>
   </category>
   <!-- The Java class that will start the Agents associated
   with the service -->
   <agentinit classname = "select.agent.test.TestInit">
   </agentinit>
</service>
<!-- The ISCN test service -->
<!-- ************************* -->
<service
   id         = "iscn"
   URI        = "http://www.iscn.com/"
   server     = "samson.aszi.sztaki.hu"
   maintainer = "micsik@sztaki.hu"
   anonymous  = "allowed">
   <description language = "en" text = "ISCN test
service"></description>
   <!-- A simple category -->
   <category
      id         = "contents"
      rater-type = "human"
      type       = "value">
      <description language = "en" text = "Example
category"></description>
      <valuecategory
          gui         = "slider"
          format      = "integer"
          min         = "1"
          max         = "5"
          calculation = "mean"/>
   </category>
   <!-- Imported categories -->
   <import service = "test" category = "quality"/>
</service>
```

```
</services>
```

9. Example of file structure on a SELECT server

Here is an example of a file structure for a SELECT server:

URL ---	**Content** -------
http://select/v1.0/	Repository of XML format specifications (DTDs) for SELECT version 1.
http://select/v1.0/services.dtd	XML format the list of services.
http://select/v1.0/service.dtd	XML format for the description of one SELECT service.
http://select/v1.0/send-rating.dtd	XML format for the send-rating operation.
http://select/v1.0/send-rating-respons e.dtd	XML format for the responses of the send-rating operation.
http://select/v1.0/evaluate-query.dtd	XML format for the evaluate query request.
http://select/v1.0/evaluate-response.d td	XML format for the response of the evaluate operation.
http://select/v1.0/select-service-desc riptions.xml	List of SELECT services in version 1 of SELECT, see 0 The SELECT general service description on page 71.
http://select/v1.0/general/	Version 1 of the select general service.
http://select/v1.0/general/common -service-description.xml	Description of common descriptors to several SELECT services.

URL ---	Content -------
http://select/v1.0/general/general-service-description.xml	Description of the general SELECT service. The general service is for everyone, not for specialised groups.
http://select/v1.0/general/id/input-ratings	Entry point for incoming non-anonymous (registered or pseudonymous) ratings to the select general service.
http://select/v1.0/general/ano/input-ratings	Entry point for incoming anonymous ratings to the select general service.
http://select/v1.0/general/search?query=	Entry point for the web-based search operation.
http://select/v1.0/general/evaluator	Entry point for the evaluate operation.
http://select/v1.0/iscn/	Version 1 of the select ISCN service.
http://select/v1.0/iscn/iscn-service-description.xml	Description of the special SELECT service for ISCN.

10. Issues for further study

These issues are items which are not needed for the base system implementations, and which may be modified by experience from the first implementation efforts.

1. The Advanced Search facility should be specified, based on query by example, SQL or some other standard query language methodology.

2. The format of the personal interest profile, and keywords is not ready. In particular, should there be a split between profile set by the user him/herself and set by automatic methods, such

as ML algorithms on the user's rating and behaviour. Also, to what extent should this profile be specified in a formal, logical language, like "If newsgroup is alt.culture.sweden then do not filter away anything", etc. In the first implementations, we will just use a simple set of unordered keywords as the personal profile.

3. Is security enough? Do we need more security features? If so, which and how?

4. Is there a need for NNTP versions of some or all of the operations?

5. Is a streaming version needed for the evaluate operation?

6. Privacy and security issues for Get-Atomic-Ratings.

7. Is more needed for ML support?

8. Is more needed for NLP support?

9. Exchange-Ratings-Data not ready. No great priority.

10. Set-Service-Description not ready. No great priority. Can be done using local or web-based interface.

11. Is more needed for thesauri support?

12. Should the SELECT protocols be based on SOAP in order to base it on something existing? SOAP is described at http://msdn.microsoft.com/xml/general/soap_white_paper.asp (quick intrduction), and http://www.develop.com/soap/ (useful links).

11. The SELECT Agent protocol

The advanced SELECT platform now supports an agent architecture that makes it easy to integrate collaborative or information filtering algorithms. The agents can run in the Select server or on a client machine over the Network and can perform tasks as maintaining datastructures that can speed-up collaborative filtering algorithms, perform collaborative filtering, notifying other agents of a change in the Select database or using machine learning at the client to derive useful user-profile information. The agents can communicate with one another and with the Select server using an extension of the original XML Select protocol. They can request certain services to be executed or can send a notification about the occurrence of a certain event.

The introduction of agent means that the protocol had to be extended to support this. This appendix gives an overview of the necessary extensions.

11.1.1 Entry points

Following new entry points in the Select server have been made:

Entry **-----**	**Function** **--------**
/agent/register-agent.xml	Register an agent with the AgentList
/agent/deregister-agent.xml	Deregister an agent with the AgentList
/agent/list-agents.xml	Request a remote AgentList

In addition to this, every remote agent and the server have the following entry points:

Entry **-----**	**Function** **--------**
/agent/request/"name"	Request an agent a service "name" is the name of the agent
/agent/notify/"name"	Notify an agent of an event. "name" is the name of the agent

11.1.2 XML Extensions

This is an overview of the new XML document type definitions.
Registering an agent with the server-side AgentList.

Register Agent Request (register-agent.dtd)	
<!ELEMENT register-agent (notification*)> <!ATTLIST register-agent id CDATA #REQUIRED URL CDATA #REQUIRED parameters CDATA #REQUIRED	name of the agent network location for notifications and requests description of the request parameters (not used) name of the notification type of notification depends on the type e.g. if

> <!ELEMENT notification EMPTY> <!ATTLIST notification id CDATA #REQUIRED type (new-rate-in-category | new-rate-by-user | changing-profile | update-in-category | alarm) 'alarm' typedata CDATA #REQUIRED >	type is "update-in-category" it’s a string of format "service,category". If type is "changing-profile" it’s the raterid of the user whose profile is monitored

Register Agent Reply (register-agent-reply.dtd)	
<!ELEMENT register-agent-reply EMPTY> <!ATTLIST register-agent-reply ok (yes | no) 'no' >	Positive or negative outcome of the register operation

Deregister an agent with the server-side AgentList

Deregister Agent Request (deregister-agent.dtd)	
<!ELEMENT deregister-agent EMPTY> <!ATTLIST deregister-agent id CDATA #REQUIRED >	Name of the agent to deregister

Deregister Agent Reply (deregister-agent-reply.dtd)	
<!ELEMENT deregister-agent-reply EMPTY> <!ATTLIST deregister-agent-reply ok (yes | no) 'no'	Possible or negative outcome of the deregister operation

>	

Get a remote AgentList for use in a remote agent

List Agents Request (list-agents-request.dtd)	
<!ELEMENT list-agents-request (session+)>	A list of one or more session identifiers
<!ELEMENT session EMPTY>	
<!ATTLIST session	Identifier of a session
id CDATA #REQUIRED	
>	

List Agents Reply (list-agents.dtd)	
<!ELEMENT list-agents (agent*)>	A list of data for one or more agents
<!ELEMENT agent EMPTY>	
<!ATTLIST agent	
id CDATA #REQUIRED	name of the agent
URL CDATA #REQUIRED	network location
parameters CDATA #REQUIRED	description of the request parameters (not used)
session-id CDATA #REQUIRED	session-id of the owner of the agent
>	

Notify a remote agent of the occurrence of a certain event

Notify Agent Request (notify-agent.dtd)	
<!ELEMENT notify-agent EMPTY>	
<!ATTLIST notify-agent	
agent-id CDATA #REQUIRED	name of the agent
notify-id CDATA #REQUIRED	name of the notification
data CDATA #REQUIRED	data of the notification (depends on type)

>	

Request an agent for some service

Agent Request (agent-request.dtd)	
<!ELEMENT agent-request EMPTY> <!ATTLIST agent-request id CDATA #REQUIRED request CDATA #REQUIRED parameters CDATA #REQUIRED >	name of the agent request string (depends on type of agent) request parameters (depends on type of agent) e.g. request = "get-n-closest-neighbors" parameters = "raterid"

Agent Reply (agent-reply.dtd)	
<!ELEMENT agent-reply EMPTY> <!ATTLIST agent-reply ok (yes | no) 'no' data CDATA #REQUIRED >	Agent request was successful or not if ok is "yes" then data contains the requested data (depends on type of agent)

11.1.3 Privacy and security policy for agents

When dealing with an architecture where agents can run on the Select server or client machines and can communicate and request services of one another, it is necessary to have a system in place that limits the access to the agents in order to avoid abuse of machine resources (server or client) or the exposure of personal profile or other database information to unauthorized persons.

First of all, remote agents can only be registered and controlled by users that are registered with the Select Server. The user first logs in and receives a session-id. This ID is associated with every agent the user registers with the AgentList. It's used for authentication in a number of cases:

- A remote agent only accepts requests if the requesting agent knows this agent’s session id.

- When a remote agent requests a remote AgentList, it must include the session-id(s) of the agents it wants access to. This way, the server only returns information about agents for which the remote agent has proven it’s authority of access.

Local agents can be "private" or "public". Private local agents do not accept requests from remote agents, only from other local agents. Public local agents can accept requests from remote agents. Both local agent types can themselves request information from remote agents.

- A public local agent only accepts request when the session-id associated with the request is present in the server. This means that the session that created the agent has not ended yet.

12. Protocol Implemtation Status

Here we provide an overview of the implementation status of the SELECT protocol as done by the SELECT EU project.

Get-Service-Description-List

Compelety implemented

Get-Service-Description

Completely implemented

Send-Rating

Completely implemented exept for the

rater-competence
rater-trust
message-id

fields of the rating. These are not used anywhere.

Set-Profile

Implemented except for

- the pseudonymous related thing. Pseudonyms are never used in the
- present system.

- Multiple profiles per user. Every user has 1 profile for all the services, it contains its unique data (name, password,…,languages spoken, reward account) and some data for the SELECT test service filtering algorithms (keywords,…)

- the reward account that is never used.

- A profile must always be replaced as a whole. Single fields cannot be changed separately.

-

Get-Profile

Completely implemented.

Login & Logout

Completely implemented?

Get Atomic Ratings

Implemented except the only the first combination of rater/URL/category is used in the lookup.

All other requested combinations are ignored.

Simple-Search Operation

Completely implemented but the search query is a regular expression. This gets matched against the keywords stored with the rated URI's and the ones with the best instant ratings are returned.

Evaluator

Implemented except the

personal
context
sort
collection-name

fields that are ignored and used nowhere

13. Security considerations

Not yet ready.

14. Copyright

15. Acknowledgments

Michel Claude, Christopher Lueg, David Mason, Andras Micsik, Massimo Vanocchi and Richard Wheeler have participated in the production of this document. An earlier attempt to encode PICS in XML format was made by O. Lassila as described in [PICS3].

16. References

Ref.	Author, title
[PICS1]	Rating Services and Rating Systems (and Their Machine Readable Descriptions) http://www.w3.org/PICS/services.html
[PICS2]	PICS Distribution Label Syntax and Communication Protocols http://www.w3.org/PICS/labels.html.
[COOKIES]	RFC 2109 HTTP State Management Mechanism. D. Kristol, L.

Montulli. February 1997.
ftp://sunic.sunet.se/rfc/rfc2109.txt

[HTTP] RFC 2068 Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee. January 1997. ftp://sunic.sunet.se/rfc/rfc2068.txt

[IMAP] IMAP4 Authentication Mechanisms. J. Myers. December 1994, RFC 1731.

[IMAP] RFC 2045-2049 Multipurpose Internet Mail Extensions (MIME). N. Freed & N. Borenstein, November 1996. ftp://sunic.sunet.se/rfc/rfc2045.txt, rfc2046.txt, rfc2047, rfc2048, rfc2049

[URI] RFC 1738 Uniform Resource Locators (URI).T. Berners-Lee et al, December 1994. ftp://sunic.sunet.se/rfc/rfc1738.txt

[XML1] Extensible Markup Language (XML) 1.0. W3C REC-xml-19980210, T. Bray, J. Paoli, C.M. Sperberg-McQueen. http://www.w4.org/TR/1998/REC-xml-19980210

[XML2] A Technical Introduction to XML, N. Walsh, Oct 1998, http://www.xml.com/xml/pub/98/10/guide0.html

[PICS3] PICS-NG Metadata Model and Label Syntax, O. Lassila, http://www.w3.org/TR/NOTE-pics-ng.metadata

[SELFUNC] SELECT, Telematics Application Programme, RE4008, Deliverable 2.1 Draft, Functional Specifications Report, by Roland Alton-Scheidl and Richard Wheeler.

[SELARCH] SELECT System Architecture, by Richard Wheeler

17. Author's Addresses

Jacob Palme
Stockholm University and KTH
Electrum 230
S-164 40 Kista, Sweden
Phone: +46-8-16 16 67
Fax: +46-8-783 08 29
Email: jpalme@dsv.su.se

Johan Kaers
S T A R L A B Research Laboratories
Sint-Michielslaan 47
B-1040 Etterbeek (Brussels), Belgium
Phone: +32-2-7400794
Fax: +32 2 7429654
Email: johan@starlab.net