A Comparison of Fulcrum SearchServer 3.7e with DOE/LSN Requirements

ISRI Staff

Technical Report 99-05
Information Science Research Institute
University of Nevada, Las Vegas

September 9, 1999

1 Introduction

The following document reviews PCDocs/Fulcrum SearchServer 3.7e Retrieval System with respect to the Department of Energy's (DOE) search and retrieval requirements for the Licensing Support Network (LSN). We used a "hands-on" evaluation to determine how closely this system meets each requirement.

1.1 Company and Product Information

Founded in 1990, PCDocs/Fulcrum is a subsidiary of PC DOCS Group International Inc. which is a publicly traded company (NASDAQ: DOCS; TSE:DXX). PCDocs claims 3500 organizations including government, professional services, manufacturing and financial services industries, employ their products worldwide with more than 700,000 end-users.

SearchServer, the PCDocs retrieval engine, is just one of the products they offer. They have several other document and information management systems that integrate and extend SearchServer's capabilities. SearchServer though can be used as a stand-alone system.

PCDocs products are tightly integrated with Microsoft Windows NT. This probably translates into new developments being available for the Windows platform first and other platforms second. This should be a consideration if this system is selected for the LSN.

SearchServer 3.7e software requirements include:

HTTP Server such as Microsoft IIS or Netscape
Frame capable browser such as Netscape or Microsoft Explorer
Perl5

2 Collection Preparation

SearchServer is a very flexible system in that almost any kind of front-end interface can be developed with relative ease. Hence, preparation of the collection prior to loading will mainly depend on decisions made during the design phase.

Documentation provided for SearchServer installation is obscure in places. However, overall installing SearchServer was a fairly quick process. Once the documentation was overcome, loading the collection was straight-forward.

Documents in the collection can be loaded using SearchSQL and the included `execsql` program, or a higher level interface for document loading could be developed. In ISRI's collection, a table with the required

columns has been created by hand (Figure 1), and then a program has been run that outputs a SearchSQL INSERT statement for each document with the appropriate columns filled in with values from the document header (Figure 2).

```
-- Create schema/table for C1 collection
CREATE SCHEMA C1 CREATE TABLE C1 (

  ------------------
  -- Built-In fields
  ------------------
  META    VARCHAR(256)  214,   -- Built-in META handling
  LINK    VARCHAR(2048) 216,   -- Built-in LINK handling
  SUMMARY VARCHAR(1024) 217,   -- Automatic summary
  HEADERS VARCHAR(1024) 300,   -- Anything between <Hn>...</Hn>

  --------------
  -- ISRI fields
  --------------
  TITLE           VARCHAR(2048),
  DOCUMENTTYPE    VARCHAR(256),
  DOCUMENTSUBTYPE VARCHAR(256),
  PUBLICATIONDATE DATE,
  AUTHOR          VARCHAR(1024),
  KEYWORDS        VARCHAR(2048),
  PAGECOUNT       INTEGER,
  DOCUMENTID      CHAR(4),
  ABSTRACT        VARCHAR(5120))

-------------------
-- Table Parameters
-------------------
PERIODIC                          -- Only reindex when told to.
STOPFILE '/opt/ss/fultext/c2.stp'
BASEPATH '/bmfd/ss/c1'
;
```

Figure 1: SearchSQL script to create a table

For our evaluation, we loaded HTML files with META tags, although this was not *required* either for loading header fields, or satisfying other requirements. See Figure 3 for the beginning of an example document.

3 Requirements Proficiency

3.1 General Requirements

(R) Year 2K Compliance. DOE-LSN is to be Year 2000 compliant.

1. Fulcrum Technologies Inc. states:
   In this statement, "Year 2000 Compliant" software means software which has the following attributes:
   - errors processing date data will not be produced in connection with the year change from December 31, 1999 to January 1, 2000;

```
INSERT INTO C1 (
  TITLE,
  DOCUMENTTYPE,
  DOCUMENTSUBTYPE,
  PUBLICATIONDATE,
  AUTHOR,
  KEYWORDS,
  PAGECOUNT,
  DOCUMENTID,
  ABSTRACT,
  FT_FLIST,
  FT_SFNAME)
VALUES (
  'Role of Colloids in Nuclear Waste Disposal',
  'Publications',
  'Conference Papers',
  DATE'1984-01-01',
  'Avogadro, A De Marsily, G',
  'Colloid Geochemistry   Radionuclide Migration  Corrosion
   Radioactive Waste Canisters  Transport Models   Backfill
   (Repository) Hydrogeochemistry  Leaching (Geochemical)
   Mathematical Models    ',
   11,
  '0100',
  'Aspects of formation and characterization of a radioactive
   colloidal fraction released by the waste form or produced by
   association with microcolloids naturally existing in ground water or
   produced either by corrosion of container material or by degradation
   of backfill material are discussed.  A filtration model has been
   developed in order to describe colloidal transport under field
   conditions. Comparison between data obtained with laboratory column
   experiments and theoretical evaluations is presented.',
  'html:s',
  '/bmfd/collections/c1/html-meta/0100.html'
);

[...]
```

Figure 2: SearchSQL script to insert documents into table

```
<html>
<head>
<TITLE>Preliminary Analysis of Geophysical Logs from the WT Series of Drill Hole
s, Yucca Mountain, Nye County, Nevada</TITLE>

<META NAME="docid" CONTENT="5069">
<META NAME="documenttype" CONTENT="Reports">
<META NAME="documentsubtype" CONTENT="Technical Reports">
<META NAME="publicationdate" CONTENT="19850000">
<META NAME="author" CONTENT="Muller, DC Kibler, JE">
<META NAME="keywords" CONTENT="Logging   Geophysical Surveys   Boreholes    Data
 Handling  Measuring Equipment And Systems    Topopah Spring Member Stratigraphi
c Correlation   Paintbrush Tuff   Crater Flat Tuff  Yucca Mountain Calico Hills
">
<META NAME="pagecount" CONTENT="60">
</head>


<body bgcolor="#ffffff">
<h2>Preliminary Analysis of Geophysical Logs from the WT Series of Drill Holes,
Yucca Mountain, Nye County, Nevada</h2>

<center>
<h2>Abstract</h2>
</center>

<p>
<Abstract>
  Geophysical logs from the WT series of drill holes correlate well
  with similar  logs  from  other  drill holes at Yucca  Mountain,
  Nevada  in the unsaturated  zone  through the same geologic units.
  The  in-situ physical properties  of  the rocks from well logs are
  consistent  with laboratory-measured  physical properties of core
  from other drill holes. The  Topopah Spring  Member  is  concluded
  to have zones that are  highly fractured  and lithophysal  in holes
  where the density and neutron logs are very  "spiky"  as noted in
  other cored drill holes.  Low levels on the uranium trace from  the
  spectral gamma-ray log indicate that fractures are neither healed
  nor filled with  materials that concentrate uranium. Therefore,
  fracture  permeability is  expected  to  be  high. This conclusion
  is  consistent  with  fracture analysis from other drill holes on
  Yucca Mountain.  The dielectric  constant and dielectric
  resistivity logs correlate well with the epithermal  neutron,
  borehole  compensated  density,  and induction  resistivity  logs
  in   the unsaturated zone.
</Abstract>
<hr>
<!-- Begin page 1 -->
```

[...]

Figure 3: HTML version of a document.

Product	**Version/ Release**	**Is Product Y2K Compliant**	**Details of Y2K Non-Compliance**	**Date by Which Product Will be Y2K Compliant and Version/Release # that Will Be Y2K Compliant**
SearchServer, SearchBuilder for C, C++, VB, Java	3.7D	Compliant w/minor issues	Implicit conversion of some 2-digit year values to 1900 base century. Some internal date limits set at 2004 and 2038.	December 31st, 1998 New Release - 3.7e
SearchServer, SearchBuilder for C, C++, VB, Java	3.7e and 4.0	Compliant		

Table 1: Fulcrum Product Y2K Compliance Table

- date-based functionality will behave consistently for the year change from December 31, 1999 to January 1, 2000;
- in all interfaces and data storage, the century in any date will be specified either explicitly or by unambiguous algorithms or interfacing rules;
- year 2000 will be recognized as a leap year;

provided that all other products (e.g. other software, firmware and hardware) used with the Fulcrum software are Year 2000 Compliant and properly exchange date data with the Fulcrum software product.
Fulcrum will not be testing releases prior to the releases stated in the Table 1. In order to obtain the next release, you must agree to subscribe to Fulcrum's Support Software service and to not let it lapse or expire. Discontinued products will not be tested.
The Fulcrum software products listed above are customizable products where user customization can be made to scripts and other shipped components. This Statement does not apply to Fulcrum software products which are customized. Customer must ensure that any customization carried out does not remove the compliance provided by Fulcrum.
The Fulcrum software products listed above utilize other third party software products and/or third party vendor network equipment that may or may not be Year 2000 Compliant.
All Fulcrum software products are provided pursuant to a software license agreement that specifies the scope of our end users' rights and contains terms and conditions governing the use of the Fulcrum software products. **This Statement does not constitute a warranty or extend the terms of any existing warranty and is merely provided in order to assist end users in determining the Year 2000 status of their Fulcrum software products and in planning for the transition to the Year 2000.**

(R) Collection Size. DOE-LSN must accommodate the anticipated size of 1,000,000+ documents containing 10,000,000+ text pages and images.

REQUIREMENT NOT TESTED

(R) Internet Accessible. DOE-LSN must be accessible on the Internet.

1. SearchServer satisfies this requirement.
2. SearchServer lacks an extensive pre-developed user interface, however, it is possible to build virtually any conceivable user interface that is desired using the supplied API and other tools. Such an interface can be designed and implemented to allow access over the Internet to the data stored in SearchServer. Since standardized Internet protocols and conventions are typically used in communications between machines across the Internet, there are virtually no restrictions on the client side for gaining access to a server running the SearchServer system.
3. There are factors that may affect accessibility of the system, all of which are considerations for any Internet communications, not just for SearchServer. These include:
   - Internet firewalls. If communications between the client and server must pass through a firewall device, it is possible that communications may be restricted due to security policies.
   - Network bandwidth. The transfer of document images and other types of data will make considerable demands on available network bandwidth. Adequate bandwidth on the server side must be available for SearchServer to serve requests from all users who wish to access the system.
   - Network latency. Latency is the apparent delay between the time some data is transmitted from one point on the network to when it is received at another point. With an interactive system such as SearchServer, it is important to consider the effect of network latency on the usability of the system. For example, if the average delay is too great between when a user clicks and a response is received, many users may feel that the system is too slow to use.
4. **Overall Impression**
   The SearchServer distribution provides the search engine, API, and other basic tools with the idea that the customer can then design and build whatever user interface is desired. A web-based user interface is one possibility that would make SearchServer's stored data accessible on the Internet.

(R) Windows/Windows NT. DOE-LSN must be usable by clients on Windows and Windows NT operating systems.

1. The SearchServer system satisfies this requirement.
2. SearchServer is usable by Windows-based clients in a number of ways, all of which depend on exactly what sort of user interface has been developed by the customer.
3. The most widely usable interface would be web-based. Windows clients can easily make use of a web-based user interface through the use of a web browser (such as Netscape Communicator or Microsoft Internet Explorer).
4. Windows-based user interfaces to SearchServer can also be built using C/C++, Visual Basic, or Java. And through the use of special coding techniques, other languages such as Perl may be used to build a Windows user interface.
5. Also, the SearchServer documentation indicates that client software is available for the Windows platform. This software was not considered in this evaluation.
6. **Overall Impression**
   As described, there are several ways that Windows systems and users can access a SearchServer database. How exactly Windows users accomplish this is entirely dependent on the SearchServer user interface, which is designed and built by the customer.

(B) Platforms/Operating System. DOE-LSN should run on one of the following platforms: Windows NT, Sun Solaris, Alpha Unix.

1. SearchServer runs on the following platforms:

Operating System	Architecture
Sun Solaris	SPARC and UltraSPARC
Microsoft Windows 95/98	Intel
Microsoft Windows NT	Intel
Bull AIX	Bull ESCALA
Hewlett Packard HP-UX	HP 9000
IBM AIX	RS/6000
Siemens SINIX-N	RM

Note: Only Sun Solaris was used for testing.

2. There are no known issues which would affect performance on this platform.

(R) Concurrent Users. DOE-LSN shall support up to 150 concurrent users. [LSS2-064]

REQUIREMENT NOT TESTED

3.2 Querying Requirements [LSS2-011]

(R) Query for Document. The DOE-LSN shall provide the capability to query the system for a list of all documents that meet the query criteria and sort the displayed list on the basis of selected displayed fields or relevancy to the query. [LSS2-011]

1. SearchServer returns a list of documents that meet the query criteria.
2. The user can sort results by document relevance or by selected fields in either ascending or descending order. Using the evaluation interface, only a primary sort field can be selected; this same field must be selected for display to submit a syntactically correct query.
3. Query for Document can be implemented as a basic function of SearchServer. In the evaluation design, a user can issue a *SearchSQL* query directly or enter query text and select options from a point-and-click interface. Figure 4 shows the querying interface designed by ISRI.
4. The options available with the ISRI interface design: Ranking, Retrieval Model, and Linguistic Processing are presented in terse terms; these could be simplified or chosen automatically based on a user's selected preferences. These options will be described in more depth in Requirement 3.2.
5. The SearchServer query language syntax, SearchSQL (described in more detail in Requirement 3.2), is an extension to SQL and would be difficult to use for an inexperienced user. A graphical user interface must be designed on top of SearchSQL. This is common practice in the applications using SQL in Relation Database Management Systems.
6. **Overall Impression**
   SearchServer has the capability to return a list of documents that satisfy a given query that are listed in order of relevance or in ascending/descending order by selected header fields. An interface to SearchSQL would be necessary for the LSN to be usable by untrained users.

(R) Query Header. The DOE-LSN shall provide the capability to query the system by specifying the content of one or more header fields to obtain a list of all documents that satisfy the query. [LSS2-011-1]

Comment: The search will be able to sort appropriate fields, such as date, accession number, etc. in ascending or descending order. It is anticipated that the DOE-LSN will allow the user to select and search multiple bibliographic header fields.

1. SearchServer has the ability to return relevant documents when queried with specific header fields or on combinations of header fields by issuing the SELECT statement. An example query is shown in Figure 1.

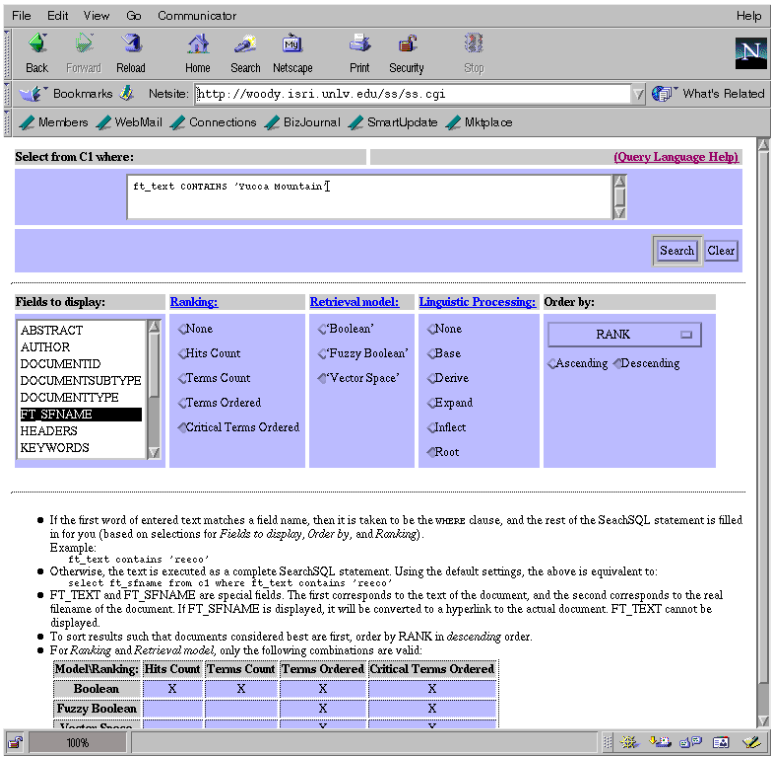

Figure 4: SearchServer Interface Evaluation Design

```
select ft_cid,relevance('V2:4') as RANK, AUTHOR FROM C1 WHERE AUTHOR CONTAINS 'Ross' OR
TITLE CONTAINS 'magnetic' ORDER BY AUTHOR DESC
```

Figure 5: Header Field Querying with SearchSQL

```
ft_text IS_ABOUT 'find documents about magnetic anomalies at Yucca Mountain'
```

Figure 6: Natural Language Query using the IS_About Predicate

```
SELECT ft_cid,relevance('V2:4') as RANK, FT_SFNAME FROM C1 WHERE FT_TEXT CONTAINS
                    'magnetic' | 'anomalies' ORDER BY RANK DESC
```

Figure 7: Structured Query using Boolean Operators

2. SearchServer provides relational operators that allow range value searching in date fields, etc.; equality is used by default.
3. Results retrieved can be ordered by relevance or in ascending or descending order as specified by the user.
4. **Overall Impression**
   SearchServer provides header field searching. A simplified interface would be a requisite to make querying usable for the LSN.

(R) Query Text. The DOE-LSN shall provide the capability to query the system by specifying one or more character strings in the full text of the document to obtain a list of all documents that satisfy the query. [LSS2-011-2]

Comment: Describe any query optimization techniques used by your system.

1. SearchServer provides the capability to specify character strings to search the full text of the document collection. Text searching is an integral part of the SearchServer system.
2. SearchServer provides two methods for querying document text:
   (a) By entering a natural language query using the IS_ABOUT predicate.
   (b) By entering a structured query with SearchServer's SearchSQL query language.
   Figures 6 and 7 show the natural language and structured language queries respectively.
3. Note that both queries require a relational statement. An interface can be designed to simplify both types of query submission. The structured query would necessitate a more complicated interface to encompass all the syntax of the language.
4. The IS_ABOUT predicate provides a way to search without having to use Boolean operators. It also allows the user to execute an *Intuitive Search* (described in Requirement 3.2) request using a single statement.
5. SearchSQL queries are specified using Boolean operators and predicates that are an extension to standard SQL. These are described in Requirement 3.2.
6. SearchServer lists several optimization techniques that will improve query response time. These should be evaluated by the system administrator to determine which index building techniques would prove most effective for the LSN's retrieval objectives.
7. **Overall Impression**
   SearchServer does not furnish a default interface for querying. A query interface as well as a simplified syntax must be developed for general use. Using SearchServer's API, a suitable interface can be implemented.

(B) Text Query Parameters. The DOE-LSN shall provide the capability to specify single and multiple character wildcards, to utilize proximity searching, and root searching as part of a full-text query and to combine multiple result sets. [LSS2-011-3]

1. SearchServer provides the query parameters listed, standard boolean type operators such as AND, OR, and NOT and has several other predicates (described in Table 2). Operators and predicates can only be used in the SearchSQL SELECT statement. These operators and predicates are an integral part of the SearchSQL. Figure 7 is an example of a SearchSQL query in SearchServer.

SearchSQL Query Language Parameters & Predicates		
Parameter	**Usage**	**Effect**
Single Character Wildcard	_ (underscore)	Matches exactly one character in the position indicated.
Right-Truncated Wildcard	% (percent sign)	Matches 0 or more characters at the end of the word specified (No left-truncation). String wildcards are ignored for Intuitive Searching.
Predicate	**Usage**	**Effect**
CONTAINS	CONTAINS 'search terms'	Indicates words or phrases that the user is trying to locate.
LIKE	LIKE 'search terms'	Used to search for a single pattern.
Boolean AND	& (ampersand)	AND operator *within* the `CONTAINS` predicate. Requires both "AND'd" terms to be present in the document.
Boolean OR	| (vertical bar)	OR operator *within* the `CONTAINS` predicate. Requires at least one of the "OR'd" terms to be present in the document.
Boolean NOT	~ (tilde)	NOT operator *within* the `CONTAINS` predicate. Requires that term not be present in the document.
Predicate AND	AND	Used in WHERE predicate. Requires both "AND'd" CONTAINS clauses to satisfy the query.
Predicate OR	OR	Used in WHERE predicate. Requires either of `CONTAINS` clauses to satisfy the query.
Predicate NOT	NOT	Excludes documents that satisfy the `CONTAINS` clause.
WITHIN	WITHIN <number_of_words>	`WITHIN` predicate is used inside a `CONTAINS` predicate. Finds documents containing terms within <number_of_words>.
PROXIMITY	PROXIMITY <number_or_words>	Used inside a `CONTAINS` predicate. It tests for proximity in the same manner as `WITHIN` but for search term lists. Finds documents containing any combination of terms specified in the lists within <number_of_words>.
BETWEEN	BETWEEN <value> AND <value>	Searches for values within thee specified range of numeric or date values.
COMPARISON	>, <, = <>, <=, >=	Used primarily for comparing numerical and date values.
IN	IN ('term1', 'term2', 'term3')	Used to select documents with terms found in a given list of exact values.

Table 2: Query Parameters and Predicates in SearchSQL

2. SearchSQL has two forms of proximity searching. The predicates: `WITHIN` and `PROXIMITY`. As described in Table 2, both predicates perform the same function but `PROXIMITY` tests for the proximity of multiple search term lists. `PROXIMITY` evaluates to 'TRUE' whenever any term in one list is within the specified distance of any term in any other list expanding the usefulness of a proximity type search.
3. SearchSQL accommodates root searching as an element of its *Linguistic Processing* capability. A complete explanation of this processing method is described in Requirement *Query Assistance* below.
4. SearchServer provides the ability to combine the results of one or more previous searches with a new search to obtain a new result using *Back Referencing*. A SearchServer API call initiates back referencing and therefore, must be integrated in a such a way that the user can perform back referencing from the user interface.
5. **Overall Impression**
   SearchServer's query language, SearchSQL, provides an extensive set of operators and predicates for searching full text. A good understanding of both the syntax and the semantics is required to construct meaningful expressions using SearchSQL. It is highly recommended that a complete, easy-to-use interface be constructed for the end user if this system is selected.

(R) Query Header and Text. The DOE-LSN shall provide the capability to query the system by specifying a combination of header field values and the text query parameters from the full text of the document to obtain a list of all documents that satisfy the query. [LSS2-011-4]

1. SearchServer is capable of querying header fields and document text in a single query.
2. As indicated, SearchSQL's syntax would be considered too complex for general use so a suitable interface should be designed to provide a simplified means of querying multiple header fields and document text in a single query.
3. **Overall Impression**
   Since header field names must be specified in the text of the query, the application programmer should design a consistent but simplified interface for combining header and text field searches.

(R) Provide Query Status. The DOE-LSN shall provide the user an indication of the query status during a query and allow the user to terminate queries in process without terminating the session or losing previous result sets. [LSS2-011-5]

> **Comment:** It is always possible to construct a query so broad that it results in an unmanageable results list. Users should be able to determine that an ongoing query is too-broad and terminate the query in process. An indication that the session is still connected and that the query is working is adequate.

1. SearchServer does not provide any specific query status indicator while the system is searching other than what is displayed by the browser.
2. The browser indicator is specific to the browser (i.e. Active "`N`" icon in Netscape Navigator, and spinning "`e`" icon in Internet Explorer) while the request is in progress. The status line at the bottom of the browser gives information about page download in bytes retrieved.
3. The download of any page transfer can be terminated by selecting the "stop" icon on the browser, but again, this is a browser feature and may differ from browser to browser. This does not interrupt the current session and does not affect any *previously saved* result sets.
4. If the user is familiar with the browser in use, then the browser indicator and the "stop" icon can aid in query status and query termination.

Linguistic Processing Methods	
Inflect	Changes the form of a base word to indicate features such as number, person, mood, or tense.
Base	Reduces inflected forms to the common base form.
Derive	Expands the root to all its morphologically possible derivations that are related, regardless of the part of speech.
Root	Reduces the word to its underived root.
Expand	Generates additional terms by removing slashes, hyphens, and parentheses.
Spell	Returns suggested words if the input word is not found in the database. A numerical parameter can be used to limit the number of alternatives.
Compound	Generates components of a compound word. (Only supported for German and Swedish.)

Table 3: Linguistic Processing Methods for Intuitive Search in SearchServer

5. SearchServer does provide the total number of documents retrieved for a query.
6. **Overall Impression**
   Most users will be familiar with their browser and will know how to observe and use these features. The user can easily determine whether the query is still in progress or stalled, but there is no other query status information.

(B) Query Assistance. The DOE-LSN shall provide interactive capabilities to assist the user in retrieving documents when the field values that uniquely define the documents are not known to the user. [LSS2-011-6]

Comment: Examples might include synonym processing, thesaurus, natural language queries, or other search aids. Because a variety of approaches are used in the commercial market, no one approach is specified.

1. SearchServer satisfies this requirement by including the following capabilities:

   Natural Language SearchServer provides a natural language query method called *Intuitive Searching*. Intuitive Searching is performed using the `IS_ABOUT` predicate which applies *linguistic processing* by default to the text argument. Natural language queries in SearchServer can be quite long so the system provides the ability to specify either a literal string, a filename, or reference a document using this predicate.

   Note: String wildcards in search terms are ignored for Intuitive Searching.

   Linguistic processing Linguistic processing is invoked by default using the `IS_ABOUT` predicate (described above). For structured queries, linguistic processing can be initiated by applying the `THESAURUS` function to individual terms or by setting the `SET_TERM_GENERATOR` with the appropriate processing methods (see Table 3) for all subsequent queries. For structured queries, linguistic processing is not the default. SearchServer's linguistic processing capability lets you expand the range of words searched by providing additional word forms. These include: inflections, derivations, spelling correction, base forms and roots as defined in Table 3. In our experiments we have used *Derive* linguistic processing to improve search results.

   Thesaurus This function invokes thesaurus expansion for a specified word or phrase. If a thesaurus is installed, it will be used to expand search terms. If no thesaurus exists, the `THESAURUS` function only initiates linguistic processing as described above. No default thesaurus is provided with SearchServer but several thesauri can be made

available to the user for a particular collection. There are six options that can be specified with the `THESAURUS` function:

WORD_SYNONYM expands the word or phrase to include equivalent words before processing the predicate.

WORD_SUFFIX expands the word or phrase to include their plural and possessive forms.

WORD_SIMILARITY applies a combination of the WORD_SYNONYM and WORD_SUFFIX options. This option gives synonym processing priority over the suffix processing. If there is a synonym match, there is no further search for an additional suffix match. However, if there is no synonym match, then suffix processing is performed.

WORD_BROADEN is equivalent to the WORD_SYNONYM option. It is included for clarity if the thesaurus file specified in the function is intended to broaden the term specified.

WORD_NARROW is equivalent to the WORD_SYNONYM option. It is included for clarity if the thesaurus file specified in the function is intended to narrow the term specified.

WORD_MODIFY uses the linguistic processing filter to expand the terms.

Relevance Feedback Using the `IS_ABOUT` predicate, SearchServer offers an ability *resembling* relevance feedback. When a set of documents from a query is returned, the user can select a portion of text from one (or more) documents and submit that text as a query itself. In our simple interface, this feature is implemented as a text box in a frame with relevant document text appearing in an adjacent frame (See Figure 8). Text can then be selected and pasted into this text box and used as a query.

Although this procedure is similar to relevance feedback in spirit, there is nothing in the documentation that suggests that some special relevance feedback algorithm is being applied. According to the documentation, a relevance feedback query with IS_ABOUT will be processed by SearchServer as any ordinary natural language query.

Retrieval Model SearchServer includes three *retrieval models* that can affect search results: 1) Strict Boolean, 2) Fuzzy Boolean, 3) Vector Space. The retrieval model can be specified as a component of the query although it is recommended that retrieval model selection be invisible to the user.

Strict Boolean This retrieval model offers very precise control over what is retrieved, but can often eliminate useful text that does not conform exactly to the search criteria. This retrieval model is the default for all predicates other than `IS_ABOUT`. It uses strict Boolean operations for combining search terms. There is no measurement of the degree to which a match is found. The use of Strict Boolean operators in a user interface is recommended only for specially trained user communities.

Fuzzy Boolean Fuzzy Boolean offers a more relaxed interpretation of the Boolean operators. Unlike Strict Boolean, it doesn't rule out documents that contain only some of the terms. Unlike Vector Space (below), Fuzzy Boolean distinguishes between AND (&) and OR (|) when calculating the aggregate weight of relevance.

Vector Space like Fuzzy Boolean, Vector Space relaxes the strict requirements imposed by the `AND`, `OR`, and `NOT` Boolean operators. Vector Space uses the AND (&) and OR (|) Boolean operators equivalently when combining terms for matching documents and for calculating the relevance value. If Boolean operators are used in the search, they are not distinguished. A model like Vector Space that doesn't require special training is usually more effective for most users.

In our precision and recall testing, we applied the Fuzzy Boolean model which gave the best results for the structured language queries. For natural language queries,

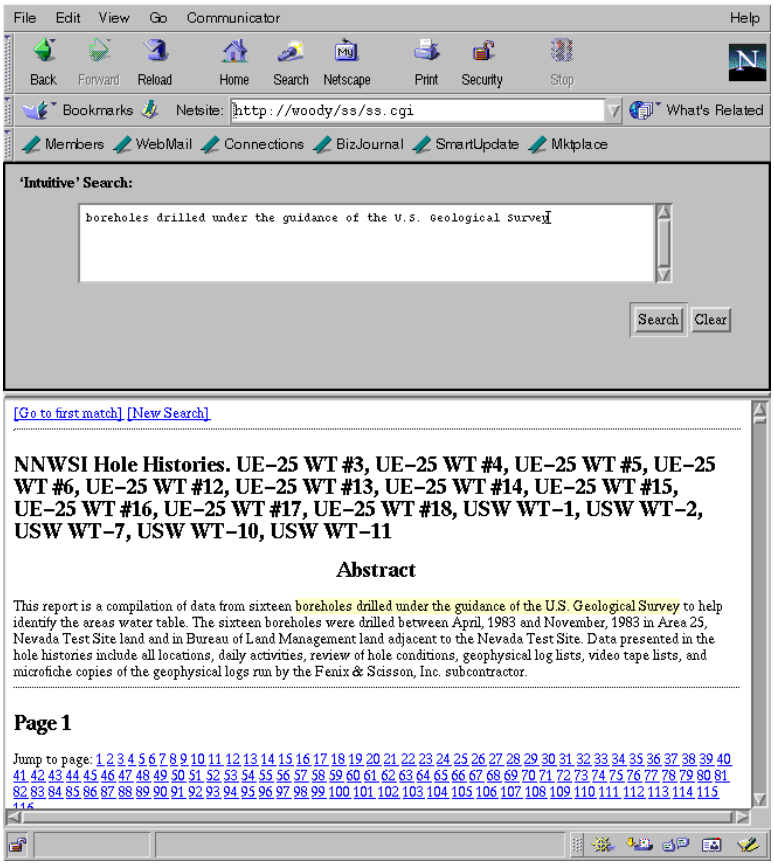

Figure 8: Simple Relevance Feedback in SearchServer

which are run using the IS_ABOUT predicate, the retrieval model is internally set to Vector Space.

Relevance Ranking Relevance Ranking combines the retrieval models (above) and ranking algorithms (listed) to provide a calculated relevance value that estimates the similarity between a document and the query criteria. The selection of a relevance algorithm determines how the relevance is measured. These algorithms use a combination of term occurrences, term frequency, inverse document frequency and user-defined search term weighting to determine document relevance.

Hits Count Counts the total number of occurrences of the individual words matched (as opposed to phrases) regardless of the term frequency in the collection.

Terms Count Counts the number of different search terms matched. The frequency of occurrence of the terms is not considered. The relevance value is the number of terms matched.

Terms Ordered Takes into account not only the number of occurrences of the search term, but also a statistical measurement of how common the term is over all documents in the collection.

Critical Terms Ordered Places emphasis on search terms that occur in fewer documents in the collection. These are the terms that are most useful in determining whether documents are relevant or not.

In our experiments as well as for the precision and recall measurements, we applied the *Critical Terms Ordered* ranking method. Our experiments on several queries and their retrieved results show that the best ranking algorithms are Terms Ordered and Critical Terms Ordered.

2. **Overall Impression**
   The only well-developed search aid available in SearchServer is the linguistic processing technique. But SearchServer offers elementary functions that can be built upon to increase system effectiveness. The basic elements are in place and its up to the system designer to implement and apply them.

3.3 Display Capabilities [LSS2-012]

NOTE: All display capabilities are dependent on the user interface built for SearchServer. SearchServer's interface can be designed and built to fit requirements. Responses to the following requirements are based on the interface built for this evaluation and SearchServer documentation.

(R) Display Document The DOE-LSN shall provide the capability to display a document. [LSS2-012]

1. SearchServer provides the ability for a user to select a document to be displayed from the results list of a query.
2. There are two ways to view the document: HTML format and raw format. The desired format can be selected by clicking on the appropriate icon on the query results page.
3. A document can only be viewed after a query has been run. No documentation was found that will provide the user the ability to view a document without running a query first.
4. Header fields are viewable. All header fields are selectable prior to query submission and are displayed in the results list.
5. The pages of each document have been combined to make a single HTML page. Page links are included for random page access. Thumbnail TIFF images for each page are embedded in the HTML with its corresponding text. Refer to Section 2 for a description of the way this collection was prepared.
6. **Overall Impression**

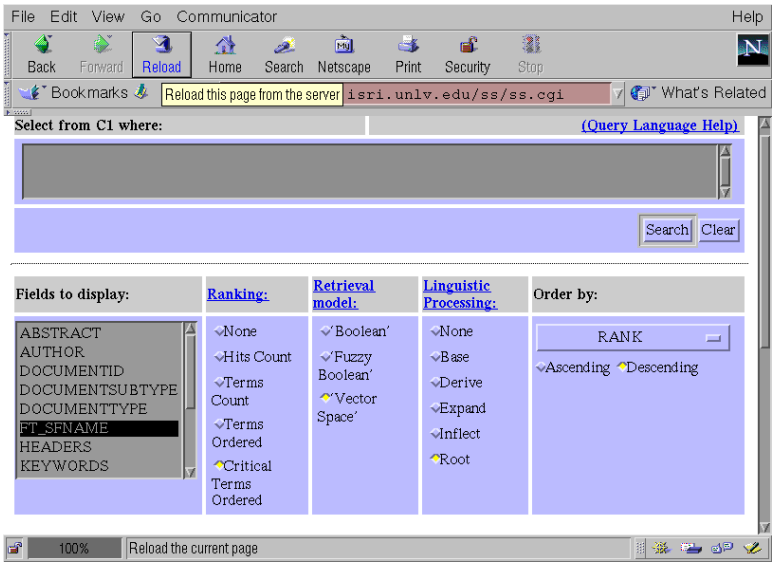

Figure 9: SearchServer Query Interface with Header Field Selection

SearchServer satisfies the document display requirement since it allows the user to view all components of a document. Since SearchServer provides flexibility in how the interface is developed, the user interface is highly customizable.

(R) Display Header The DOE-LSN shall provide the capability to display the header of a document. [LSS2-012-1]

1. SearchServer provides the user with the ability to display all or selected header fields in the query results list. These fields are pre-selected before the query is run. Any combination of header fields can be selected. Figure 9 shows the selection of header fields in the SearchServer interface.
2. **Overall Impression**
   SearchServer does provide the ability to view header fields by either selecting fields important to the user or by selecting headers. The fact that users can view header fields before selecting a document provides more information about the document and its relevance to the query.

(R) Display Text The DOE-LSN shall provide the capability to display a page of text of a document. [LSS2-012-2] Text Format: The text representation of material in DOE-LSN shall be page delimited ASCII text. [LSS2-056]

1. SearchServer displays complete document text that has been segmented into pages through the use of HTML.
2. The text of the document is placed in the browser window for viewing. The user can select pages using the links provided.
3. **Overall Impression**
   SearchServer satisfies this requirement with HTML markup added to the collection by ISRI. Other implementations of text page viewing may be possible by modifying the user interface.

(R) Locate Search Terms in Document The DOE-LSN shall provide the capability to locate the terms in the document text that satisfy a full-text query and to move from one term to the next or previous term without displaying intermediate text. [LSS2-012-3]

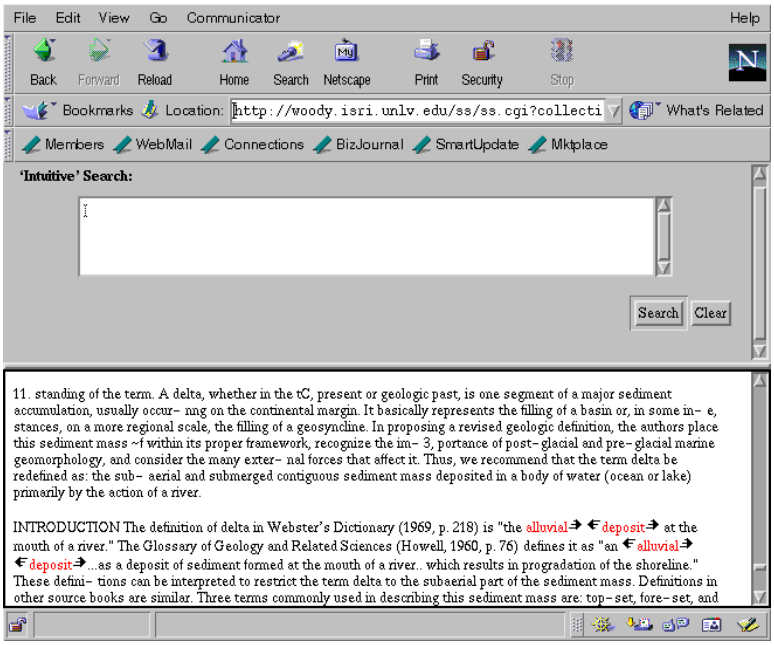

Figure 10: Highlighting Search Terms in SearchServer

Comment: This function is performed as the user is viewing the document. It is typically implemented by highlighting the search terms in the document and providing a "go to next term" function that places a cursor at the line or word of the search term.

1. SearchServer provides the user with the ability to locate search terms in the text of the document. The way in which the terms are denoted is dependent on the interface designed for the system.
2. The user interface designed by ISRI highlights search terms in a different color from the rest of the text. The ability for the user to move from one term to the next or previous term without displaying intermediate text is realized with special characters before (move to previous search term) and after (move to next search term) the term. Figure 10 shows this ability in SearchServer. This feature allows immediate location of the terms used in the query.
3. **Overall Impression**
   SearchServer provides term highlighting and the ability to move from hit-to-hit to aid in locating search terms.

(R) Display Image The DOE-LSN shall provide the capability to display the images of a document, page by page, including full page views of the images of 8-1/2 by 11 inch pages up to E-size pages.[LSS2-012-4] Image Formats: The electronic image of documentary material in DOE-LSN shall use Aldus Tagged image File Format (TIFF) Group 4 for bitonal images and Joint Photographic Experts Group (JPEG) for color and gray scale images. These formats are part of the Adobe TIFF I Version 6.0 representation. Adobe TIFF is an industry standard developed and put into the public domain by Adobe. [LSS2-057]

1. SearchServer does not directly satisfy this requirement. Only through the use of a web browser and an appropriate plugin can SearchServer satisfy this requirement. The plugin should support the TIFF image format.
2. For this evaluation, ISRI installed *AlternaTiff* a TIFF viewer plugin to meet this requirement. This plugin integrates the TIFF image directly into the browser window for viewing.

3. The user can only view the image of the document after selecting a document from the results list.
4. None of the images in our test collection were JPEG images but JPEG viewing is accommodated by most Internet browsers in use.
5. When viewing the oversized images the plug-ins will automatically fit the entire image within the viewing area of the browser.
6. **Overall Impression**
   The AlternaTIFF plugin viewer integrated into the browser by ISRI is one of the plugins satisfying this requirement. Most plugins are easy to use and their features are self-explanatory. Appropriate plugin selection is an important consideration for the LSN.

(R) Image Viewing The DOE-LSN shall provide image viewing for image enlargement, reduction, scrolling, and. [LSS2-012-5]

1. Only with the a separate TIFF viewer can SearchServer satisfy this requirement. The plugin should be capable of enlarging, reducing, scrolling and rotating the image being displayed.
2. For this evaluation, ISRI installed *AlternaTiff* a TIFF viewer plugin to meet this requirement. This plugin is equipped with buttons which, when clicked, perform the image augmentations listed.
3. **Overall Impression**
   SearchServer can meet this requirement with the appropriate plugin installed. Appropriate plugin selection is an important consideration for the LSN.

(R) Display Image and Text The DOE-LSN shall provide the capability to concurrently display an image page of a document and its text. [LSS2-012-6]

Comment: There must be a one-to-one correspondence between each page of text and its corresponding page image. This assumes each page will be tagged in the text version.

1. SearchServer can satisfy this requirement if 1) A plugin is installed which launches a new window for TIFF image display, or 2) SearchServer's user interface is designed to split the current browser into frames to display the image in one frame and the document text in the other.
2. With ISRI's implementation, TIFF image thumbnails have been embedded at the beginning of each page of text for toggling between the page and its corresponding page image. This implementation demonstrates the ability to obtain one-to-one correspondence between the text page and image page.
3. **Overall Impression**
   SearchServer's flexibility offers several methods for implementing this requirement. Some solutions, including the one used in this evaluation, may be dependent on collection preparation.

(R) Viewing Options The DOE-LSN shall allow the user to view the following combinations: 1) header, 2) image, 3) text, 4) header and text, 5) header and image, and 6) text and image.[LSS2-012-7]

1. The SearchServer interface designed by ISRI does not satisfy all display combinations listed in this requirement.
2. To view the header fields of a document the user must select these fields before running a query. The contents of these fields are listed as part of the results list.
3. The TIFF image for each document page is included as part of the HTML source. The user can click on the thumbnail to bring up the image of the current text page being viewed. The images were incorporated into the documents during collection preparation by ISRI and is not an inherent capability of SearchServer.

4. The text of the document can be displayed only after a query is run.
5. The interface designed by ISRI does not display the combinations: header and text, images and header, images and text in a single browser window. SearchServer's interface could be modified to suit these requirements.
6. **Overall Impression**
   SearchServer has the ability to satisfy this requirement. A more comprehensive interface could be built using the API provided in SearchServer.

3.4 Printing Requirements [LSS2-013]

NOTE:
The printing capabilities of SearchServer depend entirely on the browser in use. In addition, the functionality of printing and the appearance of the printed material depends on the interface designed for SearchServer. In the case of frames, the appropriate frame must be selected prior to printing. Printing capabilities were tested extensively using several versions of Microsoft Internet Explorer and Netscape but exceptions may occur.

(R) Print Document. The DOE-LSN shall provide the capability to print a document at a local printer. [LSS2-013]

Comment: It is assumed that the local printer is capable of printing the requested document type.

1. SearchServer satisfies this requirement through the use of a web browser. No special capabilities of the system are employed.
2. Exhibit A shows a printed document from the SearchServer system. It includes all document pages, image links, and text links. Other document components, like image pages, would have to be printed separately.
3. The interface devised by ISRI allows printing of both an HTML version and a raw ASCII text version. Both document formats print correctly on all browsers.
4. **Overall Impression**
   SearchServer depends on the user's browser capabilities to print a document from the DOE-LSN collection. The user needs to be familiar with frames to print successfully. The appearance of the document printout depends on the preparation of the collection (see Section 2 for more details).

(R) Print Header. The DOE-LSN shall provide the capability to print a document header at a local printer. [LSS2-013-1]

1. Printing a document header is not inherent in SearchServer.
2. The amount of detail contained in a particular header field can be selected before a query is run. This selection determines what header fields appear in the results list. With our current user interface, the header can be printed as part of the results list but not as a separate unit.
3. **Overall Impression**
   The header is associated with the results list as opposed to a selectable element for viewing. This feature though could be incorporated into the interface as a viewable and therefore, printable unit.

(R) Print Text. The DOE-LSN shall provide a user selectable capability to print from one page to all of the text of a document, and any selected ranges of pages, at a local printer. [LSS2-013-2]

Comment: The system must be able to discern pages within a document for printing.

1. SearchServer can print all the text of a document, but there is no concept of "pages" that corresponds directly with the printed pages of an LSN document.

2. Since Search Server depends entirely on the browser for printing, only an "HTML page" can be printed which in this case, is the entire document. A range of pages can be printed in some browsers but they will not be equivalent to the pages in the document.
3. Printing is not complicated to use provided that the user is already familiar with frame printing as described at the beginning of this section.
4. **Overall Impression**
   SearchServer is able to print a document in its entirety. Printing single pages or ranges of pages of text could be implemented but should be considered carefully to ensure its usefulness in the browser/HTML environment.

(B) Report Generation. The DOE-LSN should provide report generation capabilities for several of the above listed tasks.

1. SearchServer does not supply a report generation facility.
2. SearchServer's versatility would allow a report generation function to be added but would be dependent on platform statistics and would require complete implementation.
3. **Overall Impression**
   Although SearchServer does not provide this capability, it can be implemented in the system. This capability should be added during collection preparation.

(R) Print Standard Image. The DOE-LSN shall provide a user selectable capability to print from one to all images, and any selected ranges of images, of 8-1/2 by 11-inch (or smaller) pages of a document, at a local printer, reduced to a single 8-1/2 by 11-inch paper. This includes the capability of printing an oversized page image , up to E-sized, on a single 8-1/2 by 11-inch sheet of paper. [LSS2-013-3]

1. SearchServer depends on the browsers capabilities as well as an appropriate plug-in to satisfy this requirement. This capability is not an integral part of SearchServer.
2. Printing images, in particular TIFF images, may not be an inherent component of the default browser in use. In many cases, the user will need to download a plugin before they can view or print a standard image.
3. Most viewer plugins have the ability to print the current image to a local printer. Viewers *usually* include the ability to reduce, enlarge, and rotate an image page, so these capabilities should be met in general.
4. The ability to print a *selected range* or *all* page images of a document is usually not a feature of viewer plug-ins and was not built into the structure of the HTML documents in our test collection.
5. **Overall Impression**
   Apart from modifying SearchServer's interface, a plug-in viewer may need to be adapted. Selection and distribution of an appropriate plugin together with thoughtful integration of image printing into the user interface would meet this requirement.

(R) Print Oversized Image. The DOE-LSN shall provide the capability to print an oversized page image, up to E-sized, on a single sheet of paper at 100 percent of the size of the original image. [LSS2-013-4]

1. SearchServer depends on the viewer plugin and printer capabilities to satisfy this requirement. This capability is not an integral part of SearchServer.
2. Printing an oversized image depends on the capabilities of the printer being used.
3. **Overall Impression**
   Since all printing in SearchServer is browser and plugin dependent, printing oversized images depends on an appropriate plugin. Selection and recommendation of the plugin to LSN users is an important consideration.

(R) Print Results List. DOE-LSN shall provide the capability to print some or all of the summary lines of a results list. [LSS2-013-5]

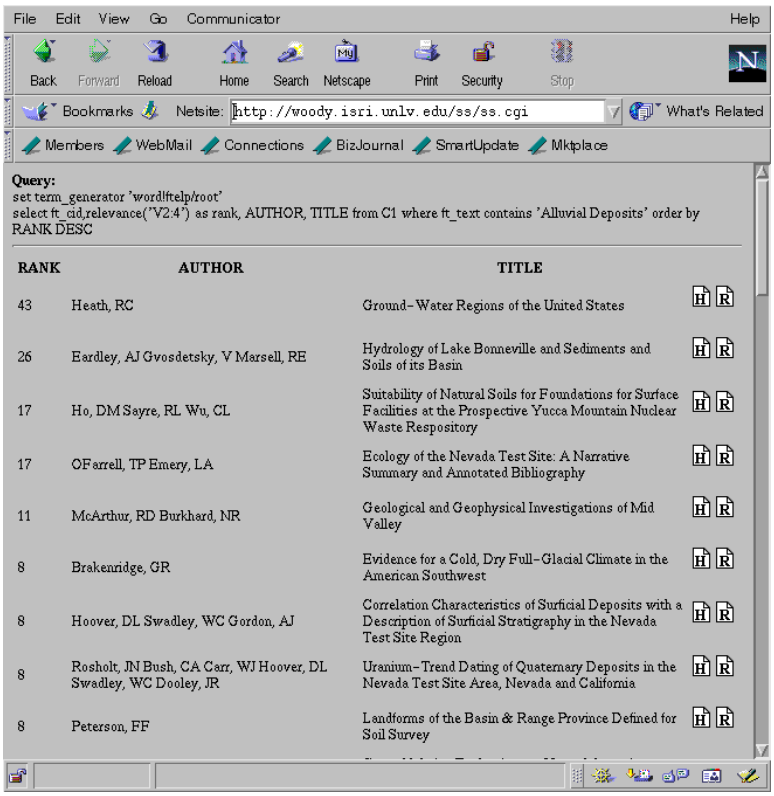

Figure 11: SearchServer Results List

1. SearchServer satisfies the ability to print the results list but it does not provide summary lines for display or print.
2. The query page gives the user the ability to modify what fields are displayed in the results list and therefore, printed in the results list. Figure 11 shows the results list with `RANK`, `AUTHOR` and `TITLE` selected for display.
3. Summary lines are not an option although the header abstract field can be selected for display. Varying the number of summary lines is not a feature of this interface but SearchServer could be modified to include it.
4. **Overall Impression**
   For our interface, the SearchServer results list is dependent on the selected header fields. The amount of detail required for each hit can be modified on the query page and printed accordingly.

(R) Print Screen. DOE-LSN shall provide the capability of printing the screen display. [LSS2-013-6]

1. Using the print capabilities of the browser, SearchServer can print the "HTML page" displayed. This may or may not define the "screen display." If the "screen" is defined as just the viewable area within the browser, printing just this portion is not possible unless the full HTML page is in view.
2. Our interface implementation makes use of frames which may cause differing behavior for screen printing depending on the currently viewable screen.
3. From a window with no frames, the current page is printed (which may or may not be equivalent to the viewable screen).
4. In frames mode, the selected frame can be printed but this will not be equivalent to a Print Screen. Each frame though can be selected for print separately.

5. **Overall Impression**
   SearchServer's capability of printing the screen depends on the HTML page being displayed. Frames in particular can only be printed separately.

(R) Request Paper Copy. DOE-LSN shall provide the capability to submit an electronic request for a paper copy of the header, images, or text of a document or of an entire results set, including oversized and color images. [LSS2-014]

1. This requirement's evaluation is based on the ease with which it could be added to the system since typically it would not be a standard feature of any search system.
2. Since SearchServer is highly customizable, "Request Paper Copy" could be easily integrated into Search Server's interface.
3. **Overall Impression**
   SearchServer is highly customizable with the special needs of its users in mind. "Request Paper Copy" is one of those special features that could be implemented using the system's API.

(R) Process Paper Copy Requests. DOE-LSN shall provide the capability to receive and read an electronic request for a paper copy of a document and print the requested copy.

> **Comment:** This is not anticipated to be a highly automated function. The requested body must be able to receive requests and print out the requested document. The rest of this function may be procedurally implemented.

1. This requirement is the receiving end of the "Request Paper Copy" requirement. Again, it would not be a standard feature in most search systems. In evaluating this requirement, we have assessed its ease of implementation.
2. Due to the adaptable nature of SearchServer, this feature can be easily integrated into the user interface.
3. **Overall Impression**
   While this is not a standard feature, SearchServer provides an API for its implementation. Building a custom interface is not only a quality of the SearchServer system it is a requirement.

3.5 System Administration Requirements

(R) Monitor System Status. DOE-LSN shall provide authorized users the capability to monitor the status of the system and communication components and to interrupt, restrict, or disable capabilities in order to optimize use of system resources. [LSS2-033]

1. This requirement is marginally satisfied by SearchServer.
2. SearchServer provides the administrative user with the ability to validate and verify a wide assortment of system variables and settings through the SearchSQL interface. The administrator can optimize indexing and searching, but there is no built-in ability to interrupt a search-in-progress. This feature could be implemented through a server side extension to the SearchServer product, but the level of effort would be non-trivial.
3. SearchServer does not provide information about current system resource utilization or current user activity.
4. **Overall Impression**
   SearchServer has a highly customizable interface that would allow software developers to implement a system satisfying this requirement. It should be said that SearchServer does not *inherently* satisfy this requirement but does provide the *ability* to satisfy it.

(R) Monitor Session Activity. DOE-LSN shall provide the capability for an authorized user to monitor user session activity levels and identify and cancel queries or other system activities. [LSS2-033-1]

1. This requirement is not satisfied by the SearchServer system.
2. The SearchServer system maintains no concept of a *user*. Any user level session control would have to be implemented through a combination of the SearchBuilder API and external program libraries.
3. The SearchServer does not provide any ability to monitor individual user sessions or to cancel user queries.
4. **Overall Impression**
   Once again, the flexible SearchServer system (and SearchBuilder API) makes the implementation of a system satisfying this requirement possible. Any user authentication and monitoring system would most likely be built through functions provided by the SearchBuilder API and the DBMS.

(R) Database Administration Tools. DOE-LSN shall provide authorized users the capability to assess the availability, integrity, and performance of the databases of the the DOE-LSN, including those pertaining to the storage of document header fields, text, and image data, and adjust database performance parameters or restrict or disable database features in order to optimize system performance.

1. This requirement is satisfied by the SearchServer system.
2. SearchServer provides the administrative user with the ability to monitor and optimize a wide range of system variables and settings. Through the SearchSQL interface and the SearchServer Administrator tool, the performance of indexing, storage, and searching can be finely tuned.
3. SearchServer also provides the ability to validate and verify data and database updates.
4. **Overall Impression**
   The SearchServer system allows the administrator a great deal of control over the database. Through simple SearchSQL commands, or more complicated utility programs, the SearchServer administrator has complete control over the database.

3.6 Internet Requirements

(R) Web Server Interface. DOE-LSN must interface with a Web Server for querying the system and returning query results.

1. SearchServer has the ability to satisfy this requirement.
2. The SearchServer system essentially just provides an API for the application developer to build whatever interface is required. A web-based interface is one possibility and the method with which the interface interacts with a web server for querying the system is entirely dependent on how the application design.
3. **Overall Impression**
   It is possible to build a user interface to the SearchServer system that can be accessed through a web server. Construction of such an interface, though, is solely the responsibility of the application developer. No comprehensive web-based user interface is provided with the SearchServer system.

(B) CGI and Perl5. DOE-LSN should use the CGI Standard and be accessible from the Perl5 programming language.

1. SearchServer satisfies this requirement.
2. Since the SearchServer system essentially just provides an API, it is up to the application developer to create CGI programs to implement a web-based user interface. Whether or not such programs conform to the CGI standard is entirely up to the developer.
3. An example web interface (written in Java) that uses CGI is included with the system, although its functionality is limited and it seems that it is only intended to be a starting point for developing more extensive interfaces.

4. Although there is no Perl5 support included with SearchServer, there is a freely available driver for the standard Perl5 DBI interface. `DBD::SearchServer` is built on top of the SearchBuilder for C API (included with SearchServer), and is available from normal Perl5 locations such as
`<http://www.perl.com/CPAN/authors/id/SHARI/>`. This third party software was actually used in the evaluation to build a web-based user interface to facilitate the testing of other requirements.
5. **Overall Impression**
Clearly it is possible to build programs to access the SearchServer system that conform to the CGI standard and SearchServer is accessible from Perl5 through the use of freely available third party software.

(B) ODBC/JDBC Compatibility. DOE-LSN should be accessible using Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC).

1. SearchServer satisfies this requirement.
2. Both ODBC and JDBC are supported in the SearchBuilder package. From the Overview in Chapter 1 of the SearchBuilder Text Reader Developer's guide:

   > SearchServer services text retrieval requests from applications that are compatible with the SearchServer API specification or with a subset of the Microsoft ODBC specification. You can create compatible applications using SearchBuilder developer's tools, or off-the-shelf products written for the ODBC application program interface (API).

3. And from the Overview in Chapter 1 of the SearchBuilder for Java Developer's Guide:

   > All of PC DOCS/Fulcrum's development products access the SearchServer Application Program Interface (API), which is based on industry specifications familiar to Database Management Systems (DBMS) programmers. These include both the Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) specifications.

4. **Overall Impression**
The documentation clearly indicates that ODBC and JDBC access methods are well-supported.

3.7 Timing Requirements

(R) Timing Strings. The DOE-LSN shall meet the average response times shown in Table 6. The performance shall be achieved with 15 concurrent DOE-LSN users active on the system. [LSS-065]

The SearchServer system cannot be evaluated at this time under the minimum required load of 15 users, or with the required 5 million pages of document data. However, the minimum timing requirements were analyzed with the smaller test collection of about 50,000 pages of document data.

These tests were all performed on a remote Windows NT 4.0 client machine. The client PC is a 450Mhz Pentium II running Netscape Communicator 4.5 to connect to the SearchServer server. The machines are connected via a 10Mbit LAN. Since we are not testing SearchServer under operational conditions, the load on the server is relatively low compared to its capabilities.

It should also be noted that only a very rudimentary interface to the SearchServer retrieval system has been implemented for our evaluation. Additional features may impact these timing tests.

1. Retrieval of query results list. LSS2-065-2

Query	Average response time (seconds)
INJD-T3-Q1	12.5
TEJA-T3-Q2	26.75

Table 4: Timing for Retrieval of Results List

Document	Page Count	Retrieval time (seconds)
1	12	1.5
2	70	3.0
3	25	2.0
4	92	3.5
5	58	2.5
6	86	3.5
7	53	2.0
8	20	1.4
9	77	3.0
10	3	1.0
Total:	496	23.5

Table 5: Timing for Retrieval of Document Text

- The DOE-LSN requires that the query results list be retrieved in 45 seconds for UNLV test queries INJD-T3-Q1 and TEJA-T3-Q2. Each query was made five times, the time to retrieve each result list was measured, and the average computed for each query over all trials computed. Table 4 summarizes the SearchServer average response time for each query.
- **Overall Impression**
  These average times are for the retrieval of all documents considered relevant by the SearchServer system. It is worth noting that SearchServer returned 374 relevant documents for query INJD-T3-Q1, and 844 for TEJA-T3-Q2. This accounts for the large difference in retrieval times for the two queries. Given the size of the current collection and the light load on the system, it is unknown if SearchServer will satisfy this requirement.

2. Retrieval of header fields for document identified in query results list. LSS2-065-3
   - The current SearchServer interface does not allow for the retrieval of header field information so this requirement was not tested.
3. Retrieval of text data for document identified in results list. LSS2-065-4
   - The LSN requires that the first page of text be retrieved in five seconds, and each subsequent page in one second. Under our implementation of the LSS prototype collection in the SearchServer system, the entire document is returned when selected from the query results list. Note that thumbnail images for each page in the document are returned as well as the corresponding text.
   - From a sample of 10 documents retrieved from a query results list, the time to retrieve the entire document from the SearchServer server was measured. These results are shown in Table 5.
   - **Overall Impression**
     From Table 5 we can infer that SearchServer retrieves approximately 20 pages per second. From the results of this experiment, it is believed that SearchServer will satisfy this timing requirement.
4. Retrieval of image data for documents identified in results list. LSS2-065-5
   - The DOE-LSN requires the first page image to be retrieved in ten seconds with each subsequent image retrieved in two seconds.

- The current implementation of the test collection in SearchServer does not allow for retrieval of the entire collection of page images for a document. A sample of ten pages from a document were retrieved. The average retrieval and display time was approximately three seconds per page.
- **Overall Impression**
  From this experiment, it is believed that SearchServer will satisfy the timing requirement for retrieval of the first page. It is unknown if the timing requirement for each subsequent page will be satisfied.

Requirement Identifier	**Function/Event**	**Conditions**	**Response Time (15/50 concurrent users)**
LSS2-065-2	Retrieval of query results list.	UNLV test query INJD-T3-Q1 or TEJA-T3-Q2* Database contains headers for at least 5 million pages of documents. A total of 10 documents found.	45 seconds/70 seconds
LSS2-065-3	Retrieval of header data for document identified in query results list.	Database contains headers for at least 5 million pages of documents.	5 seconds/8 seconds
LSS2-065-4	Retrieval of text data for document identified in query results list.	Database contains at least 5 million pages of documents.	First page: 5 seconds/8 seconds Each subsequent page: 1 second at Main Facility 2 seconds at supported sites
LSS2-065-5	Retrieval of image data for documents identified in query results list.	Database contains at least 5 million pages of documents.	First page: 10 seconds/15 seconds Each subsequent page: 2 seconds at site 3 seconds at other sites

Table 6: Response Time Requirements

EXHIBIT A:
Printed SearchServer Document