Service Manager IR Expert

Information Retrieval (IR) configuration, implementation and troubleshooting

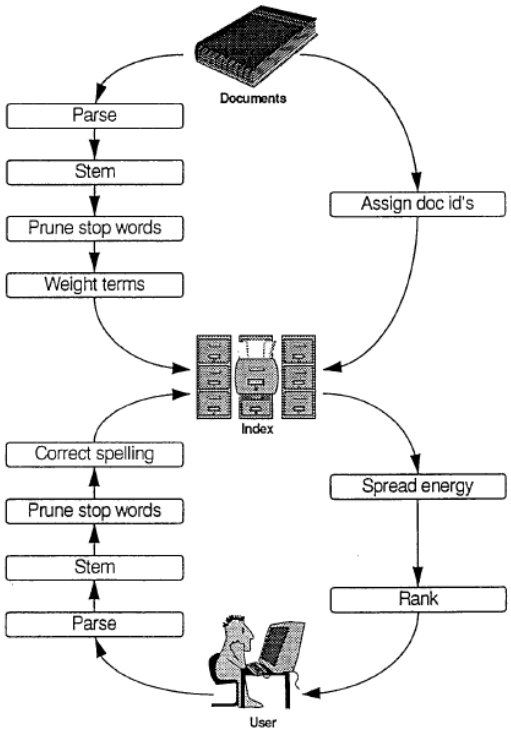
Introduction to IR Expert

Text is the primary way that human knowledge is stored, and one of the primary ways it is transmitted. Computing has changed the ways text is stored, searched, and retrieved. Automated information retrieval (IR) systems were originally developed to help manage the huge scientific literature that has developed since the 1940s. Today, with an information explosion upon us, just having a good index is not enough. Searching for words in an index is called a keyword search. Service Manager's information retrieval is not limited to search by keywords, it is closer to a natural language search as this document will illustrate.

Conceptual Model

An IR system matches user queries—formal statements of information needs—to documents stored in a database. A document is a typically textual data object.

Researchers have tried to improve IR performance by using information about the statistical distribution of terms, the frequencies with which terms occur in document collections. IR Expert uses these probabilistic models and information about term distributions, to make it possible to assign a probability of relevance to each document in a retrieved set, and allowing retrieved documents to be ranked in order of probable relevance. The following diagram outlines the information flow of IR Expert:



Document Operations

Documents are the primary objects in IR systems and support the following operations: add, delete and update. Once the documents are in the database, they need to be searched on and displayed to the end user.

Query Operations

Once a collection of documents has been indexed, queries are processed to provide a user with the desired information. Using information about term distributions, it is possible to assign a probability of relevance to each document in a retrieved set, to rank them in order of probable relevance. Queries must first be broken into their elements for lookup in the index.

Term Operations

The following term operations apply to both document indexing and query operations. Operations on terms in an IR system include lexical analysis, stemming, pruning stop words, weighting, and spelling correction.

Lexical analysis is the process of converting an input stream of characters into a stream of words or tokens. Stemming is the automated combining of related words, usually by reducing the words to a common root form. A stop list is a list of words considered to have no indexing value, used to eliminate potential indexing terms. In term weighting, terms are assigned numerical values based on the frequency with which words (or terms) occur in document collections. Since terms are often misspelled in a database, the spelling of query terms is verified and corrected if necessary to ensure the query is processed correctly. Since IR does not have a spell checker, it assumes a term was misspelled when it does not find any hits for a search and tries to find terms similar to the original search term. This occurs during the searches, IR indexes are stored in the doc as parsed.

Lexical Analysis

Lexical analysis is the process of converting an input stream of characters into a stream of words or tokens. Tokens are groups of characters with collective significance. Lexical analysis is the first stage of automatic indexing, and query processing. The lexical analysis phase during indexing produces candidate index terms that may be processed further, and eventually added to indexes. Lexical analysis of a query produces tokens that are parsed and turned into an internal representation suitable for comparison with indexes.

What is a Word?

For example, terms consisting entirely of letters should be tokens. Each token is separated from the next token by spaces. However, you have to consider the following:

- Digits – IR Expert indexes digits right along with alphabetic characters.
- Hyphens –IR Expert considers hyphenated terms as a single token and does not attempt to break the term apart.
- Other Punctuation – IR Expert allows "`", "-", and the "." to appear within a token, but not at the beginning or end of a token.
- Case –IR expert converts all terms to lower case, and is therefore case insensitive.

Stemming

To improve IR performance stemming is used to find morphological variants of search terms. If for example, a searcher enters the term crashing as part of a query it is likely that they will also be interested in such variants as crashed and crash. This process of removing the inflectional suffix of a word is called stemming.

Stemming not only allows searchers to find the variants of a term, but also reduces the size of the index files. Since a single stem typically corresponds to several full terms, by storing stems instead of the full terms, compression factors of over 50 percent can be achieved. IR Expert performs the stemming at indexing and search time.

Stemming Effectiveness

There are several criteria for judging stemmers: correctness, retrieval effectiveness, and compression performance. The effect of over-stemming on IR performance is retrieval of irrelevant documents. The effect of under-stemming on IR performance is that relevant documents will not be retrieved.

Stemming for the English and German language are included in the software, all other languages have to be localized by creating the *.stm and the *.suf files for the language.

English Stemming Algorithms

There are several types of stemming algorithms. IR Expert uses an English suffix removal stemming algorithm. Suffix removal stemming removes suffixes from terms leaving a *stem*. A simple example of such a stemmer is one that removes plurals from terms. A set of rules for such a stemmer is as follows:

- If a word ends in *ies* but not *eies* or *aies,* then change the *ies* to *y.*
- If a word ends in es but not aes, ees, or oes, then change the *es* to *e.*
- If a word ends in s, but not us or ss, then remove the s. In this algorithm only the first applicable rule is used.

The Porter Stemmer

IR Expert also uses a more advanced algorithm from Porter (1980). For more information on the porter stemmer, please refer to http://tartarus.org/~martin/PorterStemmer/def.txt or http://tartarus.org/~martin/PorterStemmer .

Pruning Stop Words

Many of the most frequently occurring words in English (e.g., the, of, and, to, etc.) are worthless as index terms. A search using one of these terms is likely to retrieve almost every item in a database regardless of its relevance. Furthermore, these words make up a large fraction of the text of most documents: the ten most frequently occurring words in English typically account for 20 to 30 percent of the tokens in a document.

To improve IR System performance, stop words are eliminated during indexing. Creating a good list of stop words includes removing all words used too frequently in the data as well as ensuring that important tokens continue to be found.

Stop list

As with lexical analysis, stop list policy will depend on the data and features of the users and the indexing process. Service Manager ships with common stop words for both the English and the German language.

To determine which terms should be added to the stop words file, run the vrir command from sm –util occasionally. Any term that occurs more than 10,000 times should be added to the stop words files. The IR system does not function reliably with terms that occur >65,000 times.

Weighting Terms

The ranking approach to retrieval allows the user to input a simple query such as a sentence or a phrase and retrieve a list of documents ranked in order of likely relevance. The natural language / ranking approach is more effective for end-users because all terms in the query are used for retrieval, with the results being ranked based on co-occurrence of query terms, as modified by statistical term weighting. This method provides results even if a query term is not the term used in the data.

Asymmetric Weights Used

IR Expert uses an inverse frequency weighting scheme. This scheme compares the frequency of a term within the current document to the frequency of the term in the collection of all documents. A term that is represented often in the current document, but rare in the collection of all documents gets the

highest weight, a term that is used often both in the current document and in the collection of all documents gets a lower weight and a term used rarely in the current document and often in the collection of all documents gets the lowest weight.

The weight going from a term to a document is

1 / DocsWithTerm

The weight going from a document to a term is

$$(\log(\text{DocsTotal} / \text{DocsWithTerm})) / \left(\sum_{f=1}^{TermsInDoc} \log(\frac{\text{DocsTotal}}{\text{DocsWithTerm}_{f}})\right)$$

where

Variable	Value
DocsTotal	Documents in entire collection
DocsWithTerm	Documents referencing a given term
TermsInDoc	Terms in a given document
f	A particular term referenced by a given document

Notice an inverse collection frequency factor of log (DocsTotal/DocsWithTerm) is used, which is large for terms that occur rarely in the collection and small for terms assigned to many documents in a collection.

This form of weighting connections between features and the documents they occur in tends to cause IR Expert to be more attracted toward those documents containing rare query terms.

Handling Spelling Errors

When handling entering text, misspelling can occur, so systems which automatically correct misspellings are a big help. For this, the system needs to recognize when an input string is significantly close to one of a set of given strings. When a user enters a query, first the lexical analysis, stemming and pruning of stop words takes place. If a word is not in the index, IR Expert attempts to identify words in the index that are close to the unrecognized word. IR Expert uses a mechanism of two successive tests.

Two-test Spelling Comparison

First, all the index terms are compared to the unrecognized word using a quick approximate matching function. Those words that are close are evaluated more precisely against the unrecognized word, taking into account the order of the letters making up the words. This second test prunes the list slowly but efficiently. Words that are very different from the unrecognized word, but happen to use similar letters, like bushland vs. husband, are eliminated at this stage. The remaining words are close to the unrecognized word.

Common Spelling Errors

Common errors on entering a word usually are:

- Adding an extra letter (e.g., *ommision)*
- Omitting a letter (e.g., differnt)
- Replacing a letter with another (e.g., *conputer)*
- Inverting letters (e.g., *comptuer)*

In most of these cases, the letter order and the characters used are unchanged, but local differences exist between the misspelled word and the correct word. Differences that cannot be explained with

any of the preceding mechanisms usually indicate that the words being compared are really different and not just misspelled.

First comparison

The first test for similarity is a comparison based on the letters present in both words. If both words use the same letters, they are deemed to have a distance of zero; otherwise their distance is the number of letters present in one word but not in the other. IR Expert only passes words to the second test that have a distance of one or less, unless changed with the ir_max_shallow_distance parameter.

This first test will return a good number of false positives (e.g. martial instead of partial), but will eliminate all irrelevant words (that is, it returns no false negatives).

Example: ServiceManager uses the letters: S E R V I C M A N G, ServiceCenter uses S E R V I C N T, so the first uses M A and G which the second does not, the second uses T which the first does not. Therefore the distance is 4. Anagrams (e.g. "ServiceManager" and "Caregivers Name") have a distance of 0.

Please note that is test only looks at letters a-z and numbers 0-5 (together 32 different characters), special characters are not taken into account so that ö and é are considered the same.

Second comparison

After a word has passed the first test, the second check compares the letter order within the words, allowing it to distinguish between, for instance, latitude and altitude. The first check would consider these words identical, whereas the second will recognize the inverting of the first two letters. This test allows IR Expert to weed out the irrelevant hits from the first test.

The main feature of this test is that words that differ because of expected mistakes will have a small distance, while very different words will have a much larger distance. For example, altitude and latitude have a distance of one, while elatitude has a distance of two from both words. Only the set of words with the lowest distance after applying test two are considered as corrections for the unrecognized word.

Spreading Activation

One of the most powerful algorithms IR Expert uses to find answers to user queries is by using spreading activation. The easiest way to learn what spreading activation does is by examining an example.

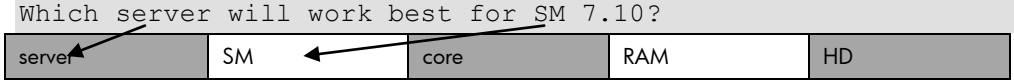
Example of Spreading Activation

Take the following query as an example:

```
Which server will work best for SM 7.10?
```

At least one term in a query must refer to an index already existing in the database. Each term mentioned in the query will be searched on.

```
Which server will work best for SM 7.10?
```



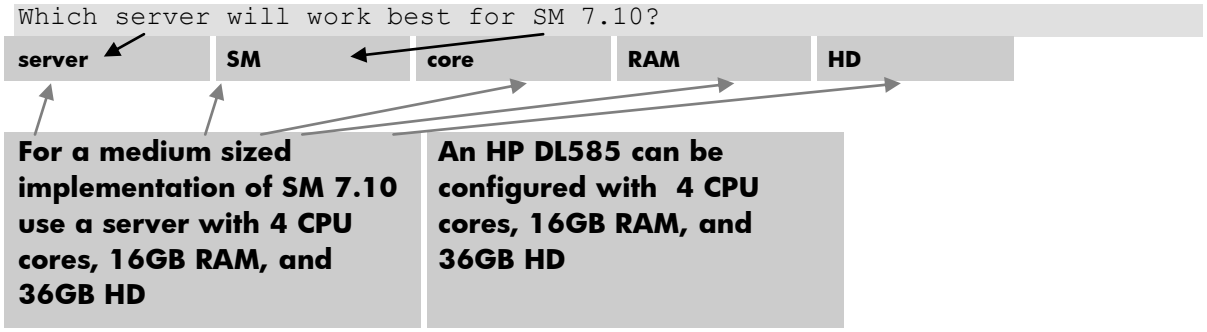
Initially; all terms found in the query are given the same weight. This weight is then propagated, first to documents that contain the active terms:

```
Which server will work best for SM 7.10?
```

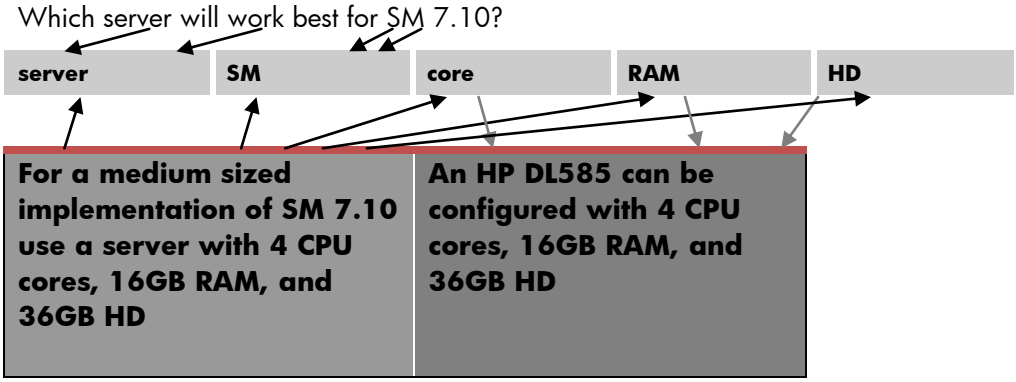
For a medium sized implementation of SM 7.10 use a server with 4 CPU cores, 16GB RAM, and 36GB HD	**An HP DL585 can be configured with 4 CPU cores, 16GB RAM, and 36GB HD**

Since the document on the left contains the term *server* and *SM* it receives energy from those terms. The amount of energy a given term like server gives to any particular document depends on the

number of documents which use that term. In this example, since only one document contains this term, all the energy from the term is sent to this one document. Activity now spreads from documents back to terms:



The amount of energy a given document spreads to a given feature depends on the statistical distribution of the feature in the document collection. Features which occur rarely will be given a greater percentage of the energy than those features which are common. Once this propagation has completed a cycle, the active features in turn spread their activation to related documents:



This sort of propagation can go on indefinitely, but in the case of IR Expert, after this point in the propagation, those documents which have reached the highest activity levels are selected as IR Expert's answer.

In this case, the document on the left would be the most active and therefore considered the best answer, while the document on the right would also become active, though not as much as the document on the left.

Comparison with Traditional Search Technologies

A traditional keyword search would miss the document on the right because the terms server or SM are not contained in the document. But the document on the right should be included as an answer to the query because of the related concept of the core, RAM and HD.

IR Expert is able to discover this relationship by means of spreading activation.

Notice too that since each document is energized to a certain degree, that by ranking these energy levels, the answer returned to the user can be listed in order of relevance. Users don't have to spend a lot of time searching through a list of equally related answers.

Shallow Matching

When entering the query “Which server will work best for SM 7.10”, IR will return the documents that contain any of the following terms: “which”, “server”, “will” or “work”, “best”, “for”, “SM”, “7.10”, unless one of them is defined as a stop word.

Deep Matching

When entering the query "Which server will work best for SM 7.10", IR will return the documents that contain any of the following terms: "which", "server", "will" or "work", "best", "for", "SM", "7.10", unless one of them is defined as a stop word. If the result set does not contain as many documents as specified in the ir_max_relevant_answers:n parameter (default: 50), deep matching uses the documents already in the list, extracts the IR keyed data, and uses that data for another IR query. For example, one document contains "For a medium sized implementation of SM 7.10 use a server with 4 CPU cores, 16GB RAM, and 36GB HD." will show up in a shallow search and in the first phase of deep match because it contains "server" and "SM" and "7.10". The second phase of deep matching will search for documents containing "for", "medium", "sized" and "implementation", etc., unless they are stop words.

Complete Matching

When entering the query "Which server will work best for SM 7.10", IR will return the documents that contain all of the following terms: "which", "server", "will" or "work", "best", "for", "SM", "7.10", unless one of them is defined as a stop word.

Cluster Analysis

Cluster analysis is a technique that assigns documents to automatically created groups based on a calculation of the degree of their association. It can be used to determine the structure of your data and has been described as a tool of discovery because it has the potential to reveal previously undetected relationships. HP does not recommend using cluster analysis in IR Expert. It is turned off by default, since it would very negatively impact IR performance.

Cluster Methods

Clustering methods are usually categorized according to the type of cluster structure they produce. IR expert uses the nonhierarchical method that divides the data into clusters; where no overlap is allowed. These are also known as partitioning methods. Each document has membership in the cluster with which it is most similar, and the cluster may be represented by a cluster representative (centroid) that is indicative of the characteristics of the items it contains.

The *hierarchical* method on the other hand produces a nested structure in which pairs of clusters are successively linked until every document is connected.

Measure of Association

In order to cluster the documents in a database, some means of quantifying the degree of association between them is required.

IR Expert determines the degree of association between any two documents by

- Considering all words in the documents only in their stemmed form
- Ignoring words containing or consisting entirely of digits
- Giving 64 points for each word occurring in both documents
- Taking away 6 points for each word contained in only one document

Cluster Algorithm

A classification system can be understood as an ordering of a series of documents or text fragments on a single conceptual shelf holding N documents. A document is denoted as D_i, with the subscript indicating the position of the document on the shelf. The documents are ordered

$D_0, D_1, \ldots, D_{N-1}, D_N$

where the subscript indicates the position of the document relative to the leftmost document on the shelf. D_0 is a hypothetical document with no subject content of any sort at the left end of the shelf.

IR Expert starts by finding the document with the least subject content and positions it at D_1. It then locates the most similar document to D_1 (using the measure of association discussed above) and positions it at D_2. This process of adding documents to the bookshelf, one at a time, continues until all the documents have been placed on the shelf.

As each document is shelved, consideration is given as to whether sufficient difference exists between the document being shelved and the last shelved document. If they are different enough and the previous cluster of documents is large enough, a partition is inserted between the two documents indicating the start of a new cluster.

After all the documents have been shelved and the partitions have been inserted, the clusters are sorted by their size, with the largest clusters first. Finally, cluster centroids are chosen to indicate the theme of each cluster.

The result is a natural grouping of the documents according to their contents. This structure allows the hidden relationships contained within a database to be seen.

Adaptive Learning

Adaptive Learning is based on two files: adlrelation and adlusermods. It requires no setup nor special rights to use.

Any Add, Update or Delete against a record in these files is processed by the server binaries and applied against the IR index. The adlrelation file is maintained through the applications.

In Service Manager 7.10 and higher, if you do not have KnowledgeManagement licensed, upon opening a Service Desk Interaction you can enter your issue description. When you click on "Find Solution" it will issue an IR query against the core file. If you then click the "Use solution" button, the application puts a record into the adlrelation file, containing the target IR file (here: "core"), the target record (the core solution used), the source file (here: "incidents"), the source record (here: e.g. "SD1001") plus the original issue description that was used as the IR query. IR takes that query from the adlrelation file and attaches it to the core record in the IR index. The core record itself is not modified, only IR's indexes to that core record.

That means if your query was "my cat ate my homework" and the core record that was used as solution only contained the term "cat", after using this solution, that same unchanged core record will be displayed if you search for "my dog ate my homework" since "homework" is now part of the IR index that points to this core record.

If the adlrelation file was modified via database manager, the IR index will be updated accordingly.

The adlusermods file works similarly but it is maintained manually through database manager. This table can be used to artificially boost the weight of a certain term for a certain IR indexed record. In the above example, you could define the core record with id 12345678 to contain "cat" 25 times, "dog" once, and "homework" 1,000 times. IR would then put the term "cat" into the IR index for that particular record 25 times, "dog" once, and "homework" 1,000 times.

Using adaptive learning will have an impact on performance. It increases the size of the IR index, which has a negative impact on IR's performance. When using the adlusermods file the index size can grow fast, so it has to be used with caution.

Adaptive Learning has an impact on the weighing. IR calculates the weight of a document separately by its real terms and by its ADL terms and then adds these numbers according to their weight. That means due to the logarithmic weight calculation that a doc with 1 "cat" in the base record and 1 "cat" in the ADL terms gets a higher weight, than a doc with 3 "cat's" in the real record.

IR Regens have been changed to take the adl files into account, watch for messages such as:

```
RTE I REGEN of file 'core' is starting
RTE I SQL scan for IR regen completed successfully. 27 records processed.
RTE I IR Regen is starting to process adlrelation file
RTE I IR Regen has finished processing 1 adlrelation records
RTE I IR Regen is starting to apply changes to IR index from shared
memory
```

Running IR in a horizontally scaled system

In Service Manager 7.11 and up, the IRQUEUE process locks an IR file (ir.*) and tries to process as many irqueue records for it as possible in a given amount of time. That amount of time can be configured with a new parameter -ir_irqueue_max_locktime:n (in seconds) and has a default value of 10. After that time period, IRQUEUE releases the lock and tries to reacquire it immediately, giving other processes a chance to take the lock and execute queries. Given this setup, in the worst case, a query might have to wait 10 seconds before it can start processing. On average, another process can wait 5 seconds while the IRQUEUE process is actively modifying IR indexes. This cycle repeats until IRQUEUE has processed all pending irqueue records. The IRQUEUE process can also be configured using the –sleep:n parameter which defines how long the IRQUEUE process will wait before looking for new records in the irqueue table. The sleep parameter accepts values in seconds and defaults to 300.

Tips and Tricks

The biggest issues customers encounter are around regenning. Using and maintaining the stop words list is highly recommended to make sure that IR queries and regens are most efficient.

Limit the amount of fields in the IR key to what is really needed. For example, an end user will be looking for a solution to a problem. At this point, he will not know the solution to the issue, but he will know the description. It does not make sense then to index the resolution field. Analyze all IR keys to ensure that only the fields are indexed that are used to find the records, not the results.

To troubleshoot IR queries, enable the ir_trace:801 parameter. This parameter can help with checking whether the query was parsed correctly, or whether the search term was replaced by a term out of the techterms table instead. In multi-byte character sets, did a Japanese character sequence get broken up into separate words, or did IR search for the complete sequence?

IR relies heavily on resource locks to protect the integrity of the IR indexes, which puts a strain on JGroups. IR is slower on horizontally scaled systems. The “Incident matching” option in the application profiles execute an IR query every time an incident is opened. Especially in horizontally scaled or larger systems, it is recommended to not use this option for performance reasons.

For best performance, we recommend that shared memory for IR should be big enough to hold all the IR indexes. To find out how big the indexes are within the scirexpert table, use the vrir option when running sm –util. IR regens do not use shared memory, but use private memory instead, so for an IR regen of a large file it may make sense to run it through sm –util, while the server is shut down, and temporarily reduce shared memory to a small value such as 24000000 so that sm –util gets more private memory.

Appendix A

IR Parameters

- ir_asynchronous

  Defines whether the Service Manager server immediately updates information retrieval files (synchronously) or whether the server creates a schedule record to process the files (asynchronously). When asynchronous IR is turned on, the sm –que:ir process has to be started to process the records from the irqueue table. If the system is horizontally scaled, asynchronous IR will be forced in Service Manager.

- ir_autostop

  When enabled, this parameter defines when IR considers a term irrelevant and stops tracking it. As an example, if ir_autostop is set to 500, when a term appears in more than 500 documents it will no longer add more documents to the term. However those 500 index entries remain in the system.

  By allowing IR to autostop terms the size of the IR files is limited and therefore the performance of IR improves. If you do not use the autostop feature then you should make sure that terms that are frequently used but of no interest in retrievals are placed in the IR stop file.

- ir_boost_same_sequence

  This parameter defines whether the Service Manager server increases the search weighting boost for documents that match the query term sequence. When enabled (value=1, default), terms used in a document that match the same sequence as terms used in the query will be considered more relevant than documents that contain the terms but are not in the same sequence.

- ir_cluster_closeness

  This parameter defines the percentage record similarity variance that records can have in an Information Retrieval search. As the percentage increases, clusters become larger and more loosely related.

- ir_cluster_symbol

  This parameter defines the alphanumeric character that indicates the system should perform a clustered query. When the system performs a clustered query, Information Retrieval searches all documents within the index to find documents with similar issues. Clustered queries are made to identify common errors.

  Important: Clustered queries require a great deal of system resources and should only be done by a knowledge expert who is trying to identify common errors. They impact IR performance negatively.

- ir_disable

  Allows you to disable the IR keys on your existing Service Manager system. This can be used during an upgrade to speed it up. At this point, setting ir_disable to 1 does not turn off IR regens as part of a system load.

  Note: After the application upgrade succeeds, you can enable the IR keys again by removing the ir_disable:1 entry from the sm.ini file and running an IR regen on all IR indexed tables.

- ir_external_files

  When set to 1, all ir.* files will be created in the ir_prefix location after a regen. We do not recommend using this option, since it makes it difficult to ensure a valid backup of the IR files. In addition, this option should not be used in a horizontally scaled system, since the IR files would not be available to all servers in the cluster.

- ir_language

  Defines the language of the text you want Information Retrieval to index. This influences which stop words file to use, how to stem the search words, etc.

- ir_languagefiles_path

Defines the path to Information Retrieval language files that contain stop words, the stem dictionary, the suffix dictionary, and the normal dictionary.

- ir_max_clusters

  Defines the maximum number of clusters to return in an Information Retrieval search.

- ir_max_deep_distance

  This parameter defines the maximum number of insertions, deletions, or substitutions that can occur in automatic spelling correction.

- ir_max_relevant_answers

  Defines the maximum number of relevant records an Information Retrieval search can return.

- ir_max_shallow_distance

  This parameter defines the maximum number of letters different a search term can be from an index term during an Information Retrieval (IR) search. The default value is 1. You can also use this parameter to turn off spelling correction.

  IR spelling correction occurs when a term used in the query is not within the IR index files. IR looks at all the terms that are in the index to determine the term that most closely matches the search term. If the number of changes required to change the query term to a known IR term is within the limits set by ir_max_shallow_distance and ir_max_deep_distance:n parameters, then the query uses the IR term. This search process is also known as a fuzzy search.

  The default value for this parameter is 1, indicating a maximum difference of 1 letter. As an example, if you search for ServiceManagre and IR Expert cannot find this term, it will allow for one letter difference and instead return ServiceManager, or ServiceManagr. It will not return documents with the word ServicManager, since it has a difference of 2 letters to the word ServiceManagre.

  Setting the parameter value to 0 disables this feature.

- ir_max_shared

  This parameter defines the maximum bytes of shared storage you want IR Expert to use. Increasing the amount of shared memory improves IR performance. Each IR file uses about 320,000 bytes of information in a hash space that is not counted by this parameter.

- ir_min_cluster_members

  This parameter defines the minimum number of records that Service Manager allows in any one cluster.

- ir_minidf

  This parameter defines the minimum relevance ranking that search terms must have for Service Manager to include them in Information Retrieval (IR) search results. Service Manager ranks each search term based on how frequently it appears in the index. The less frequent a term is in the index, the more relevance Service Manager assigns to it in search results. If a term appears too frequently in the index, then Service Manager ignores the search term as if it were in the stop word list.

  Service Manager determines a relevance ranking for each search term by computing an IDF value. Service Manager uses the following formula to compute the IDF value of search terms: [natural log (terms in index/number of instances of search term in index)]+1.

  For example, in an index of 1000 terms, a search term that appears 250 times in the index has an IDF value of 2.4. Since this is below the minimum value of 2.5, Service Manager ignores the term as too frequent. A search term that appears only 10 times in the index however has an IDF value of 5.6, and since this term exceeds the minimum IDF value threshold, Service Manager includes it in the search results.

- ir_prefix

  Defines the path to the Information Retrieval (IR) database files that contain the index if the IR index files are stored as external files.

- ir_query_drop_off

This parameter defines the maximum percentage deviation from the original search term Service Manager can use to find related records. As this percentage increases, Service Manager includes more variations of the search terms in query results.

For example, with the default 50% variance Service Manager can vary a six letter search term by three letters. Thus if the search word were cables, Service Manager could include variations such as tables and cabins in the search.

- ir_save_interval

  Defines how often Service Manager saves Information Retrieval (IR) indexes to disk. This parameter is obsolete in SM 7.11

- ir_sharedlock

  This parameter locks the shared memory allocated to Information Retrieval (IR) and prevents other Service Manager applications from accessing it. Locking IR's shared memory can improve search performance but may require additional memory for optimal performance of other Service Manager applications. This parameter is obsolete in SM 7.11.

- ir_techload

  Loads the techterms dictionary into shared memory – will always be done automatically, thus the parameter is obsolete.

- ir_term_drop_off

  This parameter defines the maximum percentage frequency that search terms can have in the Information Retrieval (IR) index for Service Manager to include them in search results. Service Manager ranks each search term based on how frequently it appears in the index. The less frequent a term is in the index, the more relevance Service Manager assigns to it in search results. If a term appears too frequently in the index, then Service Manager ignores the search term as if it were in the stop word list.

  Service Manager determines both a term search frequency and a relevance ranking for each search term. The search term frequency is a simple percentage.
  number of instances of search term in index/terms in index * 100

  The relevance ranking is determined by computing an IDF value. Service Manager uses the following formula to compute the IDF value of search terms:

  [natural log (terms in index/number of instances of search term in index)]+1

  For example, in an index of 1000 terms, a search term that appears 250 times in the index has a frequency percentage of 25% and an IDF value of 2.4. Since this is above the maximum frequency percentage value of 22%, Service Manager ignores the term as too frequent. A search term that appears only 10 times in a 1000 term index however has a frequency percentage of 1% and an IDF value of 5.6. Since this term is within the percentage frequency threshold, Service Manager includes it in the search results, although it will not be as relevant as search terms with a lower IDF value.

- ir_timelimit

  This parameter defines the maximum number of seconds that an Information Retrieval (IR) query can run. Service Manager stops queries that exceed this time limit.

Appendix B

Files involved in IR

ir.*

These files are held in the scirexpert table in Service Manager. The optional IR parameter- ir_external_files can be set to 1 if you want to store the files externally as defined by the ir_prefix parameter instead of internally in the scirexpert table. If you store the IR files externally, such as in the DATA directory, you will need to have a backup strategy in place. For example: running IR asynchronously and stopping the irqueue process prior to creating the backup and restarting it automatically once the backup program is done.

<language>.stp

Language-specific stop words file, located as defined in the ir_languagefiles_path. See the section on stop words for more information.

<language>.stm

Language specific stemming file, location is determined by the ir_languagefiles_path parameter. This file contains a set of word stems from which derivative words are formed, allowing IR Expert to match closely related words. There is no stem words file shipped with the product, but it can be created if needed. Stemming occurs based on the Porter Algorithm or English stemming as described in the stemming area of this document. Stemming is hardcoded in the product for English and German. Other languages have to be localized for IR expert by creating the *.stm and the *.suf files.

<Language>.suf

Language specific suffix file, located as defined in the ir_languagefiles_path. This file contains suffix templates used in stemming. There are no suffix files shipped with the product, but it can be created if needed. Stemming occurs based on the Porter Algorithm or English stemming as described in the stemming area of this document.

For more information

Please visit the HP Software support Web site at:

www.hp.com/go/hpsoftwaresupport

This Web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Note: Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to the following URL:

www.hp.com/go/hpsoftwaresupport/new_access_levels

To register for an HP Passport ID, go to the following URL:

www.hp.com/go/hpsoftwaresupport/passport-registration

Technology for better business outcomes