



NEMX Software Corporation

SecurExchange
Total E-mail Peace of Mind™

Intelligent Content Analysis(ICA) Concept Builder

Version 2.0

June, 2006

NEMX
SOFTWARE CORPORATION

Chapter 1: Overview 1

What is a Concept?	1
--------------------------	---

Chapter 2: Concept Policy 3

Concept Policy Terms	3
<i>Policy</i>	3
<i>Policy Items</i>	4
<i>Templates</i>	4
<i>Concept Rules</i>	4
Concept Builder Editor	5

Chapter 3: Policy Definition 7

Defining a Policy	7
Defining Templates	8
Creating Templates	8
Assigning Rules	9
Defining Policy Items	10
<i>Creating Policy Items</i>	10
<i>Assigning Templates</i>	11
<i>Assigning Restrictions</i>	12

Chapter 4: Concept Rule Reference 13

Rule Language Highlights	13
Word Stemming	14
Thesaurus Capability	14
Natural Language Query	14
Pattern Matchers	15

Definition of Query Rule Language Terms	15
<i>Query Rule:</i>	15
<i>Hit:</i>	15
<i>Search Item:</i>	15
<i>Set:</i>	16
<i>Intersection:</i>	16
<i>Intersection Quantity:</i>	16
Query Rules	16
Query Rules - At a Glance	16
Set Logic – At a Glance	17
Query rule syntax	18
Constructing a Query rule	18
Simple Keyword Search	18
Refining a Query rule	19
Adjusting Proximity Range by Specifying Delimiters	20
Using Set Logic to Weight Search Items	21
<i>Set Logic and Intersections Defined</i>	21
<i>Maximum Intersections Possible (“AND”)</i>	22
<i>Specifying Fewer Intersections</i>	23
<i>Specifying No Intersections (“OR”)</i>	24
<i>Weighting Items for Precedence (+)</i>	25
<i>Marking Items for Exclusion (“NOT”) (-)</i>	26
<i>Combinatorial Logic</i>	27
Query Rule Logic Summary	28
Query Rule Examples	29
Rule Importance to a Concept Group	29
Multiple Search Algorithms	30
REX Regular EXpression Pattern Matcher	30
Approx ApproXimate Pattern Matcher	31
Numeric Numeric Pattern Matcher	31
Single Items, Keywords, Phrases, Wildcard Patterns (*)	31
String Search (Literal Pattern Match)	32
REX Description	32
<i>REX Expression Syntax</i>	33
<i>REX Repetition Operators</i>	34
<i>Examples of Some Useful REX Expressions</i>	35
Searching for Approximations (Approx)	35
Numeric Quantities Entered as Text (Numeric)	36
Pattern Matchers Summary	38

Chapter

1

Overview

Concept Builder is a tool that allows administrators to create and manage their own email content “Concepts” and the enforcement and access to these concepts. Using Concept Builder, virtually any internal Accept Use Policy (AUP) or corporate compliance guideline can be quickly, easily and accurately represented by one or a combination of concepts. SecurExchange uses concept scanning or monitoring because it is more comprehensive and accurate than simply the use of key words, phrases and the heuristics used by other products.

What is a Concept?

When people communicate to each other they rarely use exactly the same vocabulary when trying to communicate a common idea. Typically, concepts are communicated by stringing combinations of abstract meanings together to form a concise idea. For example, if you were trying to communicate the idea of a "nice person" to someone else you might use any of the following forms:

- nice guy
- pleasing chap
- agreeable character
- excellent human being
- exceptionally fine person

While there are subtle differences in each of these phrases, the underlying concept is the same. It is also worth noting that by themselves the individual words in each phrase carry little indication of the whole idea. In a much larger sense, people string these types of concepts to form heuristically larger and more complex communications.

If you are searching for a concept within an email message body or attachment, you are actually searching for an intersection in meaning of your idea of what you are searching for with/and the body of information you are searching. SecurExchange's Intelligent Content Analysis (ICA) performs this search operation for you automatically.

Chapter

2

Concept Policy

Concepts can be combined hierarchically and are reusable, simplifying ongoing maintenance and reducing administration time and costs. Concept Builder also includes several pre-defined concepts created by Nemx, such as those for non-public personal information (NPI), solicitations, offensive content, confidential material, and others.

Concept Policy Terms

Policy

A policy is a collection of Policy Items, Templates, and Concept Rules that make up an enterprise's complete email content policy, thereby securing email content between internal and external users. The policy is applied to all email messages including attachments

Policy Items

Policy Items consist of one or more sets of Templates or Concept Rules, along with the restrictions or acceptable permissions for the transmission or receipt of those items. Policy Items also detail the actions to be applied to the message or in general as a result of the triggering of the item. A Policy Item may be a simple definition describing what makes up a piece of information, or may be quite complex by referencing existing template concepts in its definition. For instance, a complex item may be defined as requiring Concept A and Concept B, and 1 of Concept C or Concept D, but not Concept E.

Templates

A template is a re-useable Concept or a group of Concepts that can be assigned to Policy Items. Templates typically encapsulate the definition of a particular Concept into a single instance or item. Once assigned to a Policy Item or even to other templates, changes to the underlying concepts within a template are automatically propagated to all items in which this template is a member. By creating Policy Items through templates, changes to the concept of the template are quickly updated.

Concept Rules

Concept Rules are the individual aspects that make up a Concept. Each Concept Rule has a set of attributes such as search items, keywords, phrases, positioning, location, formatting, and weighting that may be defined or updated.

Concept Builder Editor

Administrators manage the policy via the Concept Builder configuration page of the SecurExchange main configuration object. The Concept Builder Editor provides a tree-table view of the complete enterprise wide policy showing all Policy Items, including their actions and restrictions, templates, and rules.

Name	Action	Applies To	Transfer Mode	R
- [icon] [checkbox checked] Policies				
+ [icon] [checkbox checked] Information Leakage				
+ [icon] [checkbox checked] Pegasus Project	Quarantine	Subject,Body,Attachment	Inbound,Outbound,Internal	
- [icon] [checkbox checked] Concept Templates				
[icon] [checkbox checked] Confidential Material				
[icon] [checkbox checked] Profanity or Coarse Language				
[icon] [checkbox checked] Racially Offensive Language				
[icon] [checkbox checked] Sexually Offensive Language				
[icon] [checkbox checked] Non Public Personal Information				
[icon] [checkbox checked] Credit Card Information				
[icon] [checkbox checked] Harassing Language				
+ [icon] [checkbox checked] SATs				

Chapter

3

Policy Definition

Defining a Policy

The typical work flow to configuring your email content policy is shown in the following diagram.

Define Templates

|

Define Policy Items

|

Apply Restrictions

|

Apply Actions

Defining Templates

Templates enable you to create "master" Concepts that can be assigned to any number of Policy Items or even other Templates. Templates are quite powerful in that they allow for the single dynamic definition of a concept of information within your email policy. As the definition of a concept changes over time, only the actual Template needs to be changed, not the numerous Policy Items that may have been initially created with or reference that template.

Creating Templates

Templates represent re-useable Concepts that can be assigned to Policy Items or other Templates. Templates are added within the Templates hierarchy level of the policy.

To create a template:

1. Add a template in one of the following ways:
 - Press the **New Template** toolbar icon
 - Right-click any entry and select the **New, Template Item**
 - Right-click the Concept Templates entry and select **Insert Group, Sub Group**.

The *New Concept Template* dialog is displayed.

2. Enter the name of the Concept

Note: The remaining settings are recommended to be left as default so that their owning Policy Item (Parent) may provide the particular settings to be used.

3. Hit Ok to save the Template

At this point the Concept Template has been created, however no aspects of the actual Concept have been defined.

Assigning Rules

Rules define the search items that make up a particular Concept. A Concept is typically made up of a number of rules. Each rule is assigned a weight or score of between 1 and 100 depending on where the query was located in the content. Different weights may be assigned based on whether the query rule was found at the top or bottom of the content, within a bullet point, subject of the message, or within a HTML link. When a rule is triggered by content within the message, including attachments, the overall score of the Concept is increased. When the combined weight of the rules within a Template or Concept Group is greater than 100 then the group is considered found or triggered.

Assign rules to a template:

1. Add a rule to a template in one of the following ways:
 - Select the Concept Template entry and press the **New Rule** toolbar icon
 - Right-click the Concept Template entry and select **Insert Rule**.

The *New Rule* dialog is displayed.

2. Enter the Query rule in terms of the words, phrases, or search items for the rule, along with delimiter bounds in which to look for the Query rule.
3. Enter the Weight when the query rule term is located within the bounds of the query rule. Different Weights may be assigned for:
 - Anywhere in the content
 - Beginning of the content
 - End of the content
 - Within the subject of the message
 - Within a HTML link
 - Within a bulleted or numbered point
4. Hit Ok to save the rule

Note: The remaining settings are recommended to be left as default so that the owning Policy Item (Parent) may provide the particular settings to be used.

Additional rules may be added to the Concept following the above procedure.

Defining Policy Items

Policy items are the true definitions that govern your active email monitoring and control aspects. Policy Items consist of templates and/or rules, which aspects of a message are analyzed, the direction a message is flowing, restrictions on the sender and/or the recipients and the resulting actions to be invoked if the Policy Item criteria is met. Any number of Policy items may be created.

Creating Policy Items

Policy Items represent the individual policy groups, their restrictions and actions that are employed on all message flow within the Exchange system. Policy Items are added within the Policy hierarchy level of the policy.

To create a policy item:

1. Add a policy item in one of the following ways:
 - Press the **New Policy Item** toolbar icon
 - Right-click any entry and select the **New, Policy Item**
 - Right-click the Policies main entry and select **Insert Group, Sub Group**.

The *New Policy Item* dialog is displayed.

2. Enter the name of the Policy
3. Select the Parent Group Inclusion for the new policy. The Parent Group Inclusion setting determines how this policy item will affect its parent in terms of being triggered.
 - Must Not Exist – The Parent will not be triggered if this Concept is found
 - Must Always Exist – The Policy Item being defined must be present for the Parent to be triggered. All of a Parents “Must Always Exists” children must be present.
 - May Exist – The Policy Item being defined may exist for a Parent item to be triggered. Only 1 of a Parent’s “May Exist” is required to be present.
 - Weighted – The Policy Item being defined contributes to the overall weight of the Parent.

Note: If this is a top level policy item, then Weighted is set by default.

4. Select the Action to be invoked if this Policy Item is triggered. Setting “Default” will result in the Action of the Parent item being used.
5. Set the appropriate options for the Transfer Mode or direction a message is traveling through the system.
 - Use Default Settings – The options for Transfer Mode are taken from the Parent Item
 - Inbound – The message must be arriving at the Exchange Server
 - Outbound – The message is leaving the Exchange Server
 - Private – The message is being transferred to/from an internal mailbox
 - Public – The message is being transferred to a public folder
6. Select the appropriate content in which the Policy Item will be monitored.
 - Use Default Setting – The options for Applies To are taken from the Parent Item
 - Subject – Content analysis will be performed on the subject of the message
 - Message Body – Content analysis will be performed on the complete text representation of the message
 - HTML Body – Content analysis will be performed on the raw HTML source of the message
 - Attachment – Content analysis will be performed on the attachments if present within messages. Microsoft Office documents (Word, Excel, PowerPoint) and PDF formats are fully supported
7. Hit Ok to save the Policy item

At this point the Policy Item has been created, however no aspects of the actual Concept have been defined.

Assigning Templates

Previously created or Nemx provided templates may be added to Policy Items to define which Concepts of information to monitor and control.

To assign a template to a policy item:

1. Add a template to a policy item in one of the following ways:
 - Right-click the Policy Item entry and select **Insert Group | Existing Group**.

The *Template* is added to the Policy Item with the Template's default settings for Parent Inclusion, Action, Transfer Mode, Applies To, and Restrictions.

2. Double click the Template just added.
3. Override any settings that are specific to this Policy Item.
4. Hit Ok to save the rule

Additional templates may be added to the Policy Item following the above procedure.

Assigning Restrictions

A Policy Item and its specific templates or rules may need to be applied to a subset of users within your Exchange Organization. Restrictions can apply to either the sender or recipient(s) of the message. When applying restrictions to a Policy Item, Template, or Rule the sender's or recipient's membership in an Active Directory / Exchange group is the determining factor. By default Policy Items, Templates, and Rules apply to all messages.

To assign a Restriction to a policy item:

1. Double click on the Policy Item in which you want to add a Restriction.
2. Select the Restrictions tab.
3. Double click on the item you wish the Restriction to apply.
4. The Select an Active Directory / Exchange DL dialog appears
5. Select the appropriate Address Book entry to show the list of available Groups/DLs
6. Select the appropriate Group / DL
7. Press the Add button.
8. Hit Ok

Repeat the above procedure to add additional restrictions.

Chapter

4

Concept Rule Reference

Rule Language Highlights

Concept Builder allows you to define rules that search for intersections of sets of lexical items, while also performing prefix and suffix stripping. Once your target is found the question arises: what rules govern proximity of the items you wish to find? In traditional searching tools, this has been done only on a line by line basis, or by using some quantitative proximity range. SecurExchange ICA can search by an intelligent textual unit, for instance a sentence or paragraph.

The rule can specify right within its definition the delimiters of choice: i.e., it can look within a sentence, paragraph, or proximity of characters. To the degree that lexical items can be defined and located as beginning and end delimiters, the content will be located within those parameters.

Search items can be more than words or phrases. You can look for a certain percentage of proximity to an entered string, find misspelled names and typos. You can also look for numeric quantities entered as text, finding "12 thousand dollars" when searching for numbers >10,000. Search items may also be powerful regular expressions.

The SecurExchange ICA will always optimize the search operations performed so that it will minimize the amount of CPU utilization and maximize the throughput search rate. At the heart of the SecurExchange ICA lies seven of the most efficient pattern matchers there are for locating items within text. With the exception of the approximate search item locator, all of these pattern matchers use a proprietary algorithmic technique that is guaranteed to out-perform any other published pattern matching algorithm (including those described by Boyer-Moore-Gosper and Knuth-Pratt-Morris).

The SecurExchange Concept Builder Rule Language was designed so that the text analysis can get rudimentary satisfaction of result right away without needing to know much of anything. At the same time, a more complex rule and concept can be written with just a little self-training time on the advanced rule syntax possibilities. We like to say that there's nothing that can't be found with a SecurExchange ICA rule.

Providing set-logic to manipulate combinations of these search items gives the ability to search for just about anything that you might want to find in the textual representation of messages or attachments. The query rule in general can be as simple or sophisticated as the organization wishes, with the simplest query being a simple natural-language question.

Word Stemming

Stemming refers to a process of stripping a word down to its root by removing suffixes or prefixes (such as the "s" on the end of English plurals, or the "re" at the beginning of some words), and then searching for valid variations of the root. This allows SecurExchange ICA to locate the root word independent of the various forms it may take. This increases the "hit rate" without providing extensive or complex rules.

Thesaurus Capability

The SecurExchange ICA can expand query rules to look for additional words or phrases that roughly have the same meaning as the initial keyword or phrase. This simplifies rule creation and improves the success rate of concept location.

SecurExchange ICA has a vocabulary of over 250,000 word and phrase associations.

Natural Language Query

You may enter a query rule in the form of a sentence or question. SecurExchange ICA will automatically identify the important words and phrases within your query rule and remove the "noise words".

Example:

What is the state of the art in text analysis?

SecureExchange will search for:

state of the art AND text AND analysis

Pattern Matchers

Pattern matchers handle certain classes of content analysis and are optimized for a particular type of searching task or format, making the overall content analysis as fast and efficient as possible. Some of the pattern matchers provide the following capabilities:

- The word-list pattern matcher can locate any word form of an entire list of words and/or phrases.
- The regular-expression pattern matcher allows for the search for things like dates, part numbers, social security numbers, and product codes.
- The approximate pattern matcher can search for things like misspellings, typos, and names or addresses that are similar.
- The numeric/quantity pattern matcher can look for numeric values that are present in the text in almost any form and allows the search to be defined generically by their value.

Definition of Query Rule Language Terms**Query Rule:**

A Concept Query rule is the question or statement of search items to be matched in the message or attachment text within specified delimiters. A Query rule is comprised of one or more search items which can be of different types, a boundary delimiter, and an appropriate weight or importance.

Hit:

A Hit is the text SecurExchange locates in response to a query rule, whose meaning matches the Query rule to the degree specified.

Search Item:

A Search Item is a word or a special expression inside a Query rule. A word is automatically processed using certain linguistic rules. Special searches are signaled with a special character leading the item (invoking a particular pattern matcher), and are governed respectively by the rules of the pattern matcher invoked.

Set:

A Set is the group of possible strings or pattern matchers that SecurExchange's ICA will look for, as specified by the Search Item. A Set can be a list of words and word forms, a range of characters or quantities, or some other class of possible matches based on which pattern matcher SecurExchange's ICA uses to process that item.

Intersection:

A portion of text where at least one member of two Sets each is matched.

Intersection Quantity:

The number of unions of sets existing within the specified Delimiters. The maximum number of Intersections possible for any given Query rule is the maximum number of designated Sets minus one.

Hits can have varying degrees of relevance based on the number of set intersections occurring within the delimited block of text, definition of proximity bounds, and weighting of search items for inclusion or exclusion.

Intersection quantity, Delimiter bounds, and Logic weighting can be adjusted by the administrator as part of a Query rule specification.

Query Rules

Query Rules - At a Glance

If you are searching for a concept within a message body, subject, or attachment content, you are actually searching for an intersection in meaning of your idea of what you are searching for with/and the body of information you are searching within.

Within SecurExchange if you invoke the query rule:

```
What are Smart Action Triggers?
```

SecurExchange's ICA will find the individual "important" terms within your query rule. Then, it will look these terms up in a thesaurus that contains over 250,000 associations and it will expand each term to the set of things that mean approximately the same thing:

SMART: able, active, clever, gifted, handsome, impudent, intelligent, shrewd, sly, vigorous, ache, brainy, bright, burn, fashionable, hurt, sting, wise,

ACTION: act, activity, battle, behaviour, blow, combat, conflict, deed, doings, execution, exploit, feat, fight, initiative, job, motion, movement, operation, perform, plot, practice, proceeding, reaction, response, step, stir, suit, trial, worl

TRIGGER: event, cause, circumstance, happening, incident, occurrence, outcome, result, effect

After SecurExchange ICA has built these sets, it will monitor the content within messages or attachment text, based on what types of text have been assigned to the query rule, for text that has all three of the concepts present within some defined boundary (i.e.; sentence, line, paragraph, page, etc.). So, if it was instructed to monitor by sentence it would be able to retrieve the following:

```
SecurExchange uses Smart Action Trigger (SAT) to handle an
organization's unique compliance requirements.
```

Also, it is not only looking for each of the words in the lists, but it is also looking for every word-form of the words in each of the lists.

Set Logic – At a Glance

Within SecurExchange's ICA, a "set" can be any one of four different types of text data:

- The set of words or phrases that mean the same thing.
- The set of text patterns that match a regular-expression.
- The set of text patterns that are approximately the same.
- The set of quantities that are within some range.

There are three types of operations that can be used in conjunction with any set:

- Inclusion -- The set must be present.
- Exclusion -- The set must not be present.
- Permutation -- X out of Y sets must be present.

The set logic operations are performed within two boundaries:

- Line.
- Sentence.
- Paragraph
- The entire text representation
- Within X number of characters of each other

Query rule syntax

Phrases are searched for either by enclosing the query rule in double quotes, or using a hyphen instead of a space. Normally the SecurExchange ICA treats hyphens and spaces the same when searching.

In addition to using the thesaurus, it is possible to specify equivalence sets directly in the query rule. That is done with a parenthesized comma separated list, e.g. (color,red,green,blue). Spaces are significant, so don't include any that are not intended. Phrases are allowed, e.g. (high tech,state of the art).

In addition to the ordinary word queries there are a number of special pattern matchers that can be used. The most recognizable pattern matcher is REX. That is introduced with a / character, and is used to search for regular expressions.

The other special pattern-matchers are the numeric pattern matcher, and the approximate pattern matcher. The numeric pattern matcher finds numbers in text, whether they are spelled out, or as digits, and can find numbers within a range. It is introduced with the # character. The approximate pattern matcher looks for a pattern similar to the desired word, and can handle transposition, dropped/extra characters including spaces. It is introduced with %.

Constructing a Query rule

The following query rule examples use simple keywords as the search items for clarity, but keep in mind that each search item can represent an entire list of things or any of the special pattern matchers. SecurExchange ICA supports many different types of search items as outlined in the Pattern Matcher section.

Simple Keyword Search

Your search can be as simple as a single word or string. If you want references to do with smart, enter the rule as “smart”.

Example:

In this example, the Query rule would look like this:

```
smart
```

When SECUREXCHANGE'S ICA executes the search, ordinances whose bodies contain matching sentences would be retrieved. An example of a qualifying sentence would be:

```
Jenny is quite a SMART person.
```

SecurExchange's prefix and suffix stripping searching logic will also find sentences like:

```
Jenny is a SMARTY pants.
```

And this sentence:

```
Jenny said after falling off her bike, "my arms SMARTS".
```

A word entered in a SecurExchange ICA query rule locates occurrences of forms of that word in both lower and upper case, regardless of how it was entered; i.e., the default keyword search is case insensitive.

ICA also expands the query rule to look for all words and phrases that have the same meaning as the initial query rule. As "intelligent", "brainy", "wise" are all synonyms of "smart" and as such SecurExchange will also located sentences like this:

```
Jenny's is quite the INTELLIGENT person.
```

```
Jenny's INTELLIGENCE is unsurpassed.
```

In some cases dictionary and thesaurus expansion is not desired. If this is the case, then adding a tilde (~) in front of a query rule word will not invoke the concept expansion for that word.

Each matched sentence is called a hit. SecurExchange locates all such hits containing "smart" and any other "smart" word forms. There would normally be quite a few hits for a common keyword query rule like this.

Refining a Query rule

To refine a query rule, thereby further qualifying what is judged a hit; add any other keywords or concepts which should appear within the same concept grouping.

Example:

```
smart action triggers
```

Fewer hits will be retrieved than when only one search item is entered (i.e., “smart”), as you are requiring that “smart” and “action” and “triggers” to occur in the same sentence. This sentence would qualify:

```
SecurExchange uses SMART ACTION TRIGGERS (SAT) to handle an  
organizations unique compliance requirements.
```

And through dictionary/thesaurus expansion and prefix/suffix stripping would locate:

```
SecurExchange uses CLEVER TRIGGERING mechanisms to react to message  
compliance violations.
```

But would not locate the following sentence as the keywords in the query rule span a sentence boundary:

```
SecurExchange is a SMART active email control product. It can  
monitor inbound, outbound, and internal message flow. Concepts of  
information can be defined which when TRIGGERED will result in some  
type of ACTION being performed on the message.
```

You may enter as many query rule items as you wish, to qualify the hits to be found.

Adjusting Proximity Range by Specifying Delimiters

By default SecurExchange’s ICA considers the entire field to be a hit when the full query rule is located within a sentence.

If you want your search items to occur within a more tightly constrained proximity range this can be adjusted.

SecurExchange allows the query rule be delimited by either a line, sentence, paragraph, within a specified number of characters, or the complete message or attachment.

Example:

```
smart action triggers -> Delimiter: Paragraph
```

By setting the delimiter to be on a paragraph boundary, the following text would be considered a hit:

```
SecurExchange is a SMART active email control product. It can
monitor inbound, outbound, and internal message flow. Concepts of
information can be defined which when TRIGGERED will result in some
type of ACTION being performed on the message.
```

Delimiters can also be expressed as a number of characters forward and backwards from the located search items. For example:

```
smart action triggers -> Delimiter: characters (25)
```

In this example “smart”, “action” and “triggers” must occur within a window of 25 characters forwards and backwards from the first item located. This provides a tighter query rule in that the words and/or phrases must be relatively closer to each other.

Using Set Logic to Weight Search Items

Set Logic and Intersections Defined

Any search item entered in a query rule can be weighted for determination as to what qualifies as a hit.

All search items indicate to the program a set of possibilities to be found. A keyword is a set of valid derivations of that word's root. A concept set includes a list of equivalent meaning words. A special expression includes a range of strings that could be matched.

Therefore, whatever weighting applies to a search item applies to the whole set, and is referred to as “set logic”.

The most usual logic in use is “AND” logic. Where no other weighting is given, it is understood that all entered search items have equal weight, and you want each one to occur in the targeted hit.

Here is an example of a typical query rule, where no special weighting has been assigned:

```
smart action trigger
```

The query rule equally weights each item, and searches for a sentence containing “smart” and “action” and “trigger” anywhere within it, finding this sentence:

```
SecurExchange is an INTELLIGENT active email control product that  
uses SMART ACTION TRIGGERD (SAT) to INTELLIGENTLY handle an  
organization’s unique compliance requirements.
```

Only those words required to qualify the sentence as a hit are located by the program, for maximum search efficiency.

In this example, there are several occurrences of the search item “smart” (smart, intelligent). It was only necessary to locate each item once to confirm validity of the hit. Such words may be found by the search in any order.

The existence of more than one matched search item in a hit is called an intersection. Specifying two keywords in a query rule indicates you want both search items to occur, or intersect, in the sentence.

A 2 item search is common, and can be thought of as 1 intersection of 2 sets.

Example:

```
smart action
```

Where something from the concept set “smart” and something from the concept set “action” meet within a sentence, there is a hit. This default set logic finds a 1 intersection sentence:

```
SecurExchange uses CLEVER triggering mechanism to REACT to message  
compliance violations.
```

”clever” is in the “smart” concept set; “react” is in the “action” concept set.

These two sets have herein intersected, forcing the context of the set members to be relevant to the entered query rule.

Maximum Intersections Possible (“AND”)

Adding a search item dictates stricter relevance requirements. Here, a sentence has to contain 2 intersections of 3 search items to be deemed a valid hit.

Example:

```
smart action trigger
```

Such a 2 intersection search finds this hit:

```
SecurExchange uses CLEVER TRIGGERING mechanisms to REACT to message compliance violations.
```

Default intersection logic is to find the maximum number of set intersections possible in the stated query rule; that is, an “AND” search where an intersection of all search items is required.

Specifying Fewer Intersections

Here is another way to write the above 2 intersection query rule, where the number of desired intersections (2) is preceded by the at sign (@):

Example:

```
@2 Smart action trigger
```

The “@2” designation is redundant as it is understood by SecurExchange to be the default maximum number of intersections possible, but it would yield the same results.

It is possible to find different permutations of which items must occur inside a hit. Even where the maximum number of intersections possible is being sought, this is still seen as a permutation, and is referred to as permuted logic.

The meaning of permuted takes on more significance when fewer intersections of items are desired.

If you wanted only one intersection of these three items, it would create an interesting range of possibilities. You might find an intersection of any of the following combinations:

```
smart (AND) action
smart (AND) trigger
action (AND) trigger
```

Specify one intersection only (@1), while listing the 3 possible items.

Example:

```
@1 smart action trigger
```

This 1 intersection search finds the following, where any 2 occurrences from the 3 specified sets occur within the hit. Hits for a higher intersection number (@2) as shown above also appear.

```
smart (AND) action
```

```
Quarantine is a type of SMART ACTION.
```

```
smart (AND) trigger
```

```
A TRIGGER is a SMART way to handle compliance violations.
```

```
action (AND) trigger
```

```
Once a message is TRIGGERED, an ACTION will be invoked on the message.
```

The “@#” intersection quantity designation is not position dependent; it can be entered anywhere in the query rule.

Any number of intersections may be specified, provided that number does not exceed the number of intersections possible for the entered number of search items.

Specifying No Intersections (“OR”)

Using this intersection quantity model, what is commonly understood to be an “OR” search is any search which requires no (zero) intersections at all. In an “or” search, any occurrence of any item listed qualifies as a hit; the item need not intersect with any other item.

Designate an “or” search using the same intersection quantity syntax, where zero (0) indicates no intersections are required (@0):

Example:

```
@0 smart action trigger
```

In addition to the hits listed above for a higher number of intersections, the following 0 intersection hits would be found, due to the presence of only one item (a or b or c) required:

```
SecurExchange is an INTELLIGENT active email control product.
```

```
SecurExchange invokes ACTIONS on messages.
```

```
SecurExchange TRIGGERING mechanism is extremely flexible.
```

All such items are considered permuted, at intersection number zero (0).

Weighting Items for Precedence (+)

Intersection logic treats all search items as equal to each other, regardless of the number of understood or specified intersections. You can indicate precedence for a particular search item which falls outside the intersection quantity setting.

A common example is where you are interested chiefly in one subject, but you want to see occurrences of that subject in proximity to one or more of several specified choices. This would be an “or” search in conjunction with one item marked for precedence. You definitely want A, along with either B, or C, or D.

Use the plus sign (+) to mark search items for mandatory inclusion. Use @0 to signify no intersections are required of the unmarked permuted items. The number of intersections required as specified by '@#' will apply to those permuted items remaining.

Example:

```
+confidential @0smart action trigger
```

This search requires (+) the occurrence of “confidential”, which must be found in the same sentence with either “smart”, “action”, or “trigger”.

The 0 intersection designation applies only to the unmarked permuted sets. Since “confidential” is weighted with a plus (+), the “@0” designation applies to the other query rule items only.

This query rule finds the following hits:

```
+confidential (and) smart
```

```
Using set logic is a CLEVER way of locating CONFIDENTIAL information.
```

```
+confidential (and) action
```

```
Once CONFIDENTIAL information is located, a message ACTION can be invoked.
```

```
+confidential (and) trigger
```

```
An attachment containing CONFIDENTIAL data can TRIGGER an archive operation.
```

More than one search item may be marked with a plus (+) for inclusion, and any valid intersection quantity (@#) may be used to refer to the other unmarked items. Any search item, including phrases and special expressions, may be weighted for precedence in this fashion.

Marking Items for Exclusion (“NOT”) (-)

You can exclude a hit due to the presence of one or more search items. Such mandatory exclusion logic for a particular search item falls outside the intersection quantity setting, as does inclusion, and applies to the whole set in the same manner. This is sometimes thought of as “NOT” logic, designated with a minus sign (-).

A common example is where one item is very frequently used in the text, so you wish to rule out any hits where it occurs. You want an intersection of A and B, but not if C is present.

Use the minus sign (-) to mark search items for exclusion. Default or specified intersection quantities apply to items not marked with a plus (+) or minus (-). The number of intersections required will apply to the remaining permuted items.

Example:

```
smart trigger -gun
```

This search has the goal of finding smart ways of handling rule triggers. However, the presence of the word “gun” might incorrectly produce references about fire arms.

Excluding the hit if it contains “gun” retrieves these hits:

```
smart (and) trigger (not) gun
```

```
SecurExchange uses CLEVER TRIGGERING mechanisms to handle message compliance violations.
```

But excludes this hit:

```
smart (and) trigger (not) gun {Excluded Hit}
```

```
The RIFLE had a SMART TRIGGERING mechanism that would only fire when aimed at inanimate objects.
```

More than one search item may be marked with a minus (-) for exclusion, along with items marked with plus (+) for inclusion, and any valid intersection quantity (@#) specification. Any search item, including phrases and special expressions, may be marked for exclusion in this fashion.

Combinatorial Logic

Weighting search items for inclusion (+) or exclusion (-) along with an intersection specification which is less than the maximum quantity possible (the default “AND” search) can be used in any combination.

A rather complicated but precise query rule might make use of weighting for inclusion, exclusion, and also a specified intersection quantity, as follows.

Example:

```
+confidential @1 smart action trigger -gun
```

The above query rule makes these requirements:

- “confidential” must be present, plus ...
- 1 intersection of any 2 of the unmarked search items “smart”, “action”, “trigger”, but ...
- Not if “gun” is present (i.e., exclude it).

This query rule retrieves the following hits, while excluding hits containing “gun”.

```
confidential, (and) smart (or) action (but not) -gun
```

```
CONFIDENTIAL material if enabled will invoke a SMART ACTION on the message.
```

```
confidential, (and) smart (or) trigger (but not) -gun
```

```
SecurExchange's INTELLIGENT content analysis will locate INTERNAL USE ONLY material within an attachment and cause a rule TRIGGER.
```

```
confidential, (and) action (or) trigger (but not) -gun
```

```
Sending SECRET business plans will trigger SecurExchange to invoke a block type ACTION.
```

Any search item, including keywords, wildcards, concept searches, phrases, and special expressions, can be weighted for inclusion, exclusion, and combinatorial set logic.

Query Rule Logic Summary

- Logic operators apply to any entered search item.
- Precedence (mandatory inclusion) is expressed by a plus sign (+) preceding the concept or expression. A plus (+) item is “Required”.
- Exclusion (“not” logic) is expressed by a minus sign (-) preceding the concept or expression. A minus (-) item is “Excluded”.
- Equivalence (“and” logic) may be expressed by an equal sign (=) preceding the concept or expression. An equal sign is assumed where no other logic operator is assigned. An equal (=) item is “Permuted”.
- Search items not marked with (+) or (-) are considered to be equally weighted. Intersection quantity logic (@#) applies to these unmarked (=) permuted sets only.
- Where search items are not otherwise marked, the default set logic in use is “And” logic. The maximum number of intersections possible is sought.
- Designate “Or” logic with zero intersections (@0), applying to any unmarked permuted search items.
- Logic operators and intersection quantity settings can be used in combination with each other, referred to as combinatorial logic.
- Sets (or lists) of things are specified by placing the elements within parenthesis, separated by commas. Example: (bob,joe,sam,sue)

Query Rule Examples

Query Rule	Finds
bob sam joe	Bob with Sam and Joe
bob sam -joe	Bob with Sam without Joe
bob sam joe @1	Bob with Sam, or Bob with Joe, or Joe with Sam
A B C D @1	AB or AC or AD or BC or BD or CD
+A B C D @1	ABC or ABD or ACD
A B C -D @1	(AB or AC or BC) without D

The plus(+) and minus(-) operators must be attached to the term to which they apply. There must be a space between the operator and any preceding term.

Correct	Incorrect
bob +sam -joe	bob + sam - joe
	bob+sam-joe

SecurExchange ICA's use of logic should not be confused with Boolean operators. SecurExchange ICA deals with these logic operators as sets rather than single strings, a different methodology.

Rule Importance to a Concept Group

When a query rule is considered a hit, its significance contributes to the overall weight of the Concept it is a member of. Therefore the individual query rules of a Concept Group should be assigned with the appropriate importance of that rule to the overall Concept Group.

Weighting of query rules is based on the location of the hit within the message or attachment or the type of formatting surrounding the hit and the number of hits within the complete textual item. For instance a hit of “confidential” within the footer of a message attachment has much more importance or weight than if it was found within a single paragraph of the message body. However if “confidential” was located 10 times within the message body or was located within a bullet point, then its overall significance would be higher.

A query rule hit can have specific weights assigned to it based on the following locations:

Anywhere:	The hit was found at some location in the text
Beginning:	The hit was found in the first 10% of the text.
Ending:	The hit was found in the last 10% of the text.
Subject:	The hit was found in the subject of the message, or header/footer or Meta data of a Microsoft Office or PDF document
HTML Link:	The hit was located within a link to a web, ftp or external site
Bullet:	The hit was found within a bullet, numbered list, or within a single sentence paragraph.

Weights assigned to a rule hit are accumulative with the exception of Anywhere and Subject. For example if a hit was at the beginning of a message body and was a sentence by itself, the weight of that rule hit would be the total of both the Beginning and Bullet weights for the query rule.

Multiple Search Algorithms

SecurExchange’s ICA allows for several methods of locating content. You can enter a natural language question. You can specify which words, phrases, or regular expressions you wish to monitor, and your query rule will be processed accordingly. To accomplish all this, several different search algorithms are used which go about pattern matching in different ways. Other than word and phrase matching (along with prefix/suffix stripping and thesaurus expansion), the chief pattern matchers available are:

REX Regular EXpression Pattern Matcher

REX makes it possible to look for fixed or variable length regular expressions of any kind and is integrated into the SecurExchange ICA so that you can mix and match words and regular expressions. You signal REX by putting a forward slash (/) in front of the word or expression.

Approx ApproXimate Pattern Matcher

XPM allows you to specify an “almost right” pattern which you are unsure of, so that you can find approximately what you have specified. XPM is also integrated into the search procedure and can be mixed in with word searches and REX regular expressions; you signal XPM with a percent sign (%) denoting the percentage of proximity to the entered pattern you desire.

Numeric Numeric Pattern Matcher

NPM allows you to look for numeric quantities in text which may have been expressed in English. NPM does number crunching through all possible numbers found in the text to locate those numbers which are in the specified range of desired numbers. It is generally used in combination with some other search item, such as a unit. NPM is signaled with a pound sign (#) preceding the numeric quantity you wish to match.

Single Items, Keywords, Phrases, Wildcard Patterns (*)

The simplest search would be to enter one keyword, like “smart”. All sentences (or otherwise delimited hits) containing the word smart or any of its equivalences will be retrieved.

To restrict a word to that root word only, excluding any equivalences, precede the word with a tilde (~). Entering “~power” as an item will find “power”, “powerful”, “powers”, etc., but will not include the 57+ equivalences that are part of the set “power”.

Entering more than one keyword on the query line will be interpreted as 2 search items, as delimited by a space character, unless it is a phrase known by SecurExchange. To link any words together as a phrase you need only put it in quotes or separate the words of the phrase with a hyphen. For example, “smart action” or “smart-action” must find those two words in that sequence, as a phrase.

A wildcard '*' can be used along with an English word to extend a rooted pattern by up to 80 characters per asterisk '*'. For example, “smart*trigger” would locate “Smart Action Triggers”. More than one asterisk '*' may be used.

A wildcard item can be searched for in intersection with other search items as well. For example: “smart*trigger compliance” would locate the sentence “Smart Action Triggers is an excellent way of handling compliance violations.”

String Search (Literal Pattern Match)

To locate a literal string (pattern matching) enter a slash '/' preceding the string within the query rule. If you want to enter a whole line to be viewed as one string, put it in quotes, with the forward slash inside the quotes. Example:

```
"/Smart Action Trigger"
```

When you denote a slash (/), you're signaling SecurExchange's ICA to use REX, bypassing the usual English word processing that goes on. REX can sometimes be more direct when such a task is all that is required.

REX Description

REX stands for Regular EXpression Pattern Matcher. REX gives you the ability to match ranges of characters, symbols, and numbers, as well as to selectively designate how many of each you wish to look for. By combining such pattern designations into what is called a "Regular Expression" you can look for such things as phone numbers, chemical formulas, social security numbers, dates, accounting amounts, names entered in more than one form, ranges of years, text formatting patterns, and so on.

REX expressions can be entered as search items by following a forward slash (/) with the REX expression.

REX Expression Syntax

Expressions are composed of characters and operators. Operators are characters with special meaning to REX. The following characters have special meaning:

"\=+*{ }, []^\$. -!" and must be escaped with a '\' if they are meant to be taken literally. The string ">>" is also special and if it is to be matched, it should be written "\>>".

- A '\' followed by an 'R' or an 'I' mean to begin respecting or ignoring alphabetic case distinction. (Ignoring case is the default.) These switches DO NOT apply inside range brackets.
- A '\' followed by an 'L' indicates that the characters following are to be taken literally up to the next '\L'. The purpose of this operation is to remove the special meanings from characters.
- A sub-expression following '\F' (followed by) or '\P' (preceded by) can be used to root the rest of an expression to which it is tied. It means to look for the rest of the expression "as long as followed by ..." or "as long as preceded by ..." the sub-expression following the \F or \P, but the designated sub-expression will be considered excluded from the located expression itself.
- A '\' followed by one of the following 'C' language character classes matches that character class: alpha, upper, lower, digit, xdigit, alnum, space, punct, print, graph, cntrl, ascii.
- A '\' followed by one of the following special characters will assume the following meaning: n=newline, t=tab, v=vertical tab, b=backspace, r=carriage return, f=form feed, 0=the null character.
- A '\' followed by Xn or Xnn where n is a hexadecimal digit will match that character.
- A '\' followed by any single character (not one of the above) matches that character.
- The character '^' placed anywhere in an expression (except after a '[') matches the beginning of a line. (same as: \x0A in Unix or \x0D\x0A in DOS)
- The character '\$' placed anywhere in an expression matches the end of a line. (\x0D\x0A)
- The character '.' matches any character.
- A single character not having special meaning matches that character.
- A string enclosed in brackets [] matches any single character from the string. Ranges of ASCII character codes may be abbreviated as in [a-z] or [0-9]. A '^' occurring as the first character of the string will invert the meaning of the range. A literal '-' must be preceded by a '\'. The case of alphabetic characters is always respected within brackets.
- The '>>' operator in the first position of a fixed expression will force REX to use that expression as the "root" expression off which the other fixed expressions are matched.

This operator overrides one of the optimizers in REX. This operator can be quite handy if you are trying to match an expression with a `!` operator or if you are matching an item that is surrounded by other items. For example: “x+>>y+z+” would force REX to find the “y’s” first then go backwards and forwards for the leading “x’s” and trailing “z’s”.

- The `!` character in the first position of an expression means that it is NOT to match the following fixed expression. For example: “start=!finish+” would match the word “start” and anything past it up to (but not including the word “finish”). Usually operations involving the NOT operator involve knowing what direction the pattern is being matched in. In these cases the `>>` operator comes in handy. If the `>>` operator is used, it comes before the `!`. For example: “>>start=!finish+finish” would match anything that began with “start” and ended with “finish”. THE NOT OPERATOR CANNOT BE USED BY ITSELF in an expression, or as the root expression in a compound expression. NOTE: This NOT operator “nots” the whole expression rather than its sequence of characters, as in earlier versions of REX.

REX Repetition Operators

A regular expression may be followed by a repetition operator in order to indicate the number of times it may be repeated.

An expression followed by the operator “{X, Y}” indicates that from X to Y occurrences of the expression are to be located. This notation may take on several forms: “{X}” means X occurrences of the expression, “{X, }” means from X to N occurrences of the expression, and “{, Y}” means from 0 (no occurrences) to Y occurrences of the expression.

- The `?` operator is a synonym for the operation “{0, 1}”. Read as: “Zero or one occurrence.”
- The `*` operator is a synonym for the operation “{0, }”. Read as: “Zero or more occurrences.”
- The `+` operator is a synonym for the operation “{1, }”. Read as: “One or more occurrences.”
- The `=` operator is a synonym for the operation “{1}”. Read as: “One occurrence.”

Examples of Some Useful REX Expressions

To locate phone numbers:

```
1?\space?(?\digit\digit\digit)?[\-\space]?\digit{3}-\digit{4}
```

To locate social security numbers:

```
\digit{3}-\digit{2}-\digit{4}
```

To locate text between parentheses:

```
(=[^()]+)      <- without direction specification
```

or

```
>>([!]+)      <- with direction specification
```

To locate paragraphs delimited by an empty line and 5 spaces:

```
>>\n\n\space{5}=!\n\n\space{5}+\n\n\space{5}
```

To locate numbers in scientific notation; e.g., “-3.14 e -21”:

```
[+\-]?\space?\digit+\.\?\digit*\space?e?\space?[+\-]?\space?\digit+
```

You can formulate patterns of things to look for using these types of patterns. You can look for a REX expression by itself, or in proximity to another search item. Such a search could combine a REX expression in union with an intelligent concept search. For example, you could enter the following as a query rule:

```
"/\digit{2}%" measurement
```

The REX expression indicates 2 occurrences of any digit followed by a percent sign. “Measurement” will be treated as a root word with its list of equivalences. For this query rule, SecurExchange ICA will look for an intersection of both elements inside the specified delimiters, and may come up with a hit such as:

```
They estimated that only 65% of the population showed up to vote.
```

where “estimated” was associated with “measurement”, and “65%” was associated with the pattern “\digit{2}%”.

Searching for Approximations (Approx)

In any search environment there is always a fine line between relevance and irrelevance. Any configuration aims to allow just enough abstraction to find what one is looking for, but not so much that unwanted hits become distracting. Speed is also an important consideration; one does not want to look for so many possibilities that the search is overly burdened and therefore too slow in response time.

SecurExchange thoroughly handles this problem through the use of XPM, ApproXimate Pattern Matcher. The intent behind XPM is that you haven't found what you believe you should have found, and are therefore willing to accept patterns which deviate from your entered pattern by a specified percentage. The percentage entered on the query line is the percentage of proximity to the entered pattern (rather than the percent of deviation).

Let us say you are looking for something that happened in Reykjavik. You can define a query rule and use XPM for the word you don't know how to spell. For example:

```
event %64Rakechavick
```

This query will look for an intersection of the word “event” (plus all its equivalences) and a 64% approximation to the entered pattern “Rakechavick”. This will in fact successfully locate a hit which discusses “events” which occurred at “Reykjavik”.

When looking for this sort of thing, you can keep lowering the specified percentage until you find what you want. You'll notice that the lower the specified proximity, the more “noise” you allow; meaning that in this case you will allow many patterns in addition to “Reykjavik”, as you are telling the program to look for anything at all which approximates 64% of the entered pattern.

Numeric Quantities Entered as Text (Numeric)

NPM, the Numeric Pattern Matcher, is one of several pattern matchers that can be used as a search item in a query rule. It is signified by a pound sign `#' in the starting position.

There are still many numeric patterns that are best located with a REX expression to match the range of characters desired. However, when you need the program to interpret your query as a numeric quantity, use NPM. NPM does number crunching through all possible numbers found in the text to locate those numbers which are in the specified range of desired numbers.

Since all numbers in the text become items to be checked for numeric validity, one should tie down the search by specifying one or more other items along with the NPM item. For example you might enter on the query line:

```
cosmetic sales $ #>1000000
```

Such a search would locate a sentence like:

```
Income produced from lipstick brought the company $4,563,000 last year.
```

In this case “income” is located as a match to “sales”, “lipstick” is located as a match to “cosmetic”, the English character “\$” signifying “dollars” is located as a match to “\$”, and the numeric quantity represented in the text as “4,563,000” is located by NPM as a match to “#>1000000” (a number greater than one million).

Another example:

```
cosmetic sales $ #>million
```

Even though one can locate the same sentence by entering the above query, it is strongly recommended that searches entered on the query line are entered as precise numeric quantities. The true intent of NPM is to make it possible to locate and treat as a numeric value information in text which was not entered as such.

The safest way to enter NPM searches is by specifying the accurate numeric quantity desired.

Example:

```
date #>=1980<=1989
```

This query will locate lines containing a date specification and a year, where one wants only those entries from the 1980's. It would also locate dates in legal documents which are spelled out (i.e. One Thousand and Eighty Five).

Pattern Matchers Summary

Query Rule	Finds
<code>ronald %regan</code>	Ronald Raygun, Ronald Re-an, Ronald 8eagan
<code>%75MYPARTNO9045d/6a</code>	Anything within 75% of looking like MYPARTNO9045d/6a
<code>/19[789][0-9]</code>	1970-1999
<code>[1-9]{3}\-[0-9]{4}</code>	Phone numbers: 555-1212, 820-2200
<code>#87</code>	four score and seven, 87
<code>#>0<1</code>	Fractions like 9/16, 55%, 0.123, 15 nanoseconds