SearchBlox Version 4.0

USER GUIDE

March 2007

SearchBlox Software, Inc.
www.searchblox.com
info@searchblox.com

Contents

continued on next page

1- Introduction

SearchBlox Content Search Software simplifies adding search functionality on your portal, intranet or website. With a unique combination of ease of customization and simplified search software management, SearchBlox offers the most cost-effective solution to your corporate search challenges.

1.1 End User Features

- Seamlessly search across RSS and Atom Web Feeds, HTTP(S), File System and Custom content
- Advanced Search - Search by file format, language, keyword occurrence and modified date
- Spelling Suggestions - Using words from indexed content
- Date Range search - restrict search results to a particular date range
- Automatic highlighting of user search query terms in HTML and PDF documents
- Keyword-in-Context Display - search results are displayed with areas of content where the keyword occurs
- User-defined number of search results per page
- Supports Boolean AND, OR, and NOT searches, Fuzzy and fielded searches
- Browsable Categories for quick access to categorized content
- Sort - search results can be sorted by date, relevance or alphabetically
- Hit Highlighting - query terms are highlighted on content title and description
- Collections - users can limit search to specific collections

1.2 Administrator Features

- AJAX-based Admin Console - easy to use and intuitive console to manage all aspects of the Search application
- Choice of Memory-Based Index (for very fast indexing) or Disk-Based Index (for large document collections)
- Built-in Replication to synchronize search indexes across multiple instances of SearchBlox [Enterprise Edition Only]
- Collections - create up to 250 document collections with customized settings
- Look & Feel - search results customizable using XSLT stylesheets. Can also be delivered as XML
- Automatic Generation of Browsable Categories using Category metadata in feeds and documents
- Built-in Crawlers to index HTTP, HTTPS, File System, RSS and Atom Web Feed content
- Selective indexing of sections of HTML pages using <noindex> </noindex> or <!--stopindex--> <!--startindex--> tags
- Support for indexing content through Proxy Servers

- Protected Content - crawlers can index content protected with Basic HTTP and Form-Based Authentication

- Reporting - real-time reporting with weekly, daily and hourly top queries and zero match queries for up to 3 months

- On-Demand & Scheduled Indexing of content

- Addition and Deletion of individual documents from the index

- Disable stemming for individual indexes

1.3 Developer Features

Note: The features are available in SearchBlox FREE and ENTERPRISE Editions only.

- REST-API - Simple platform independent API to add and delete custom content

- Automatically parse and index documents accessible over HTTP or stored in the file system

- Override document content and/or metadata at index time

- Built-in browser-based SearchBlox Development Environment to ease development

1.4 Content Features

SUPPORTED FILE TYPES

- HTML
- Excel
- PowerPoint
- RTF
- Word
- Text
- PDF

SUPPORTED FEED FORMATS

- RSS (0.90, 0.91 Netscape, 0.91 Userland, 0.92, 0.93, 0.94, 1.0 and 2.0)
- Atom 0.3

SUPPORTED LANGUAGES

- Arabic
- Bengali
- Chinese(Simplified)
- Chinese(Traditional)
- Czech
- Danish
- Dutch
- English
- Estonian
- Finnish
- French
- German
- Greek
- Gujarati
- Hebrew
- Hindi
- Hungarian
- Italian
- Japanese
- Kannada
- Korean
- Latvian
- Lithuanian
- Malayalam
- Malayalam
- Norwegian
- Polish
- Portuguese
- Russian
- Romanian
- Slovak
- Slovenian
- Spanish
- Swedish
- Tamil
- Telug
- Thai
- Turkish

- Stopwords - separate stopword list for each supported language

- MetaTags - supports standard meta tag fields (title, description, keyword)

2 - Installation

SearchBlox J2EE Search Component is supports JDK 1.4 and 1.5. It deploys to all major Java application servers.

SearchBlox Server is an integrated application incorporating everything you need to run SearchBlox. The SearchBlox Server includes SearchBlox J2EE Component, the Jetty Application Server and the Java Runtime Environment (JRE) 1.5. With the SearchBlox Server, there are no additional software requirements to deploy SearchBlox.

2.1 Requirements

Java Version	1.4.x , 1.5
Memory Requirements	Java Virtual Memory Allocation for the application server depends on size of Index but a minimum of 256 MB.
Platforms Tested	Windows, Linux, Mac OS X, IBM AIX, HP-UX, Solaris, FreeBSD, z/OS
Application Server Requirements	Servlet 2.3 / JSP 1.2
Application Servers Tested	Apache Tomcat 4.1.30 BEA Weblogic 7.x or higher IBM WebSphere 5.x or higher Pramati 3.x or higher Caucho Resin 3.x or higher JBoss 3 or higher Jetty Macromedia JRun 4.x or higher Oracle 9iAS, 10g Sun Application Server

2.2 Deployment

For SearchBlox J2EE component deployment, please follow the instructions for the respective server.

2.2.1 JBoss

Note: This guide refers to the JBoss 3.2.2 / Tomcat 4.1.27 package.

1. Create a folder named searchblox.war in the /server/deploy directory of your chosen configuration (all, default or minimal).

2. Extract searchblox.war, using for example the Java jar tool or a regular unzip command. On Windows, applications like WinZip work as well. Extract the files into the newly created /searchblox.war directory.

3. Start JBoss.

4. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.2 Jetty

1. Create a folder named searchblox under /webapps in the jetty directory.

2. Extract searchblox.war, using for example the Java jar tool or a regular unzip command. On Windows, applications like WinZip work as well. Extract the files into the newly created /searchblox directory.

3. Edit the etc/demo.xml file to add following context:

```
<Call name="addWebApplication">
<Arg>/searchblox/*</Arg>
<Arg><SystemProperty name="jetty.home" default="."/> /webapps/
   searchblox</Arg></Call>
```

4. Start Jetty.

5. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.3 Macromedia JRun

1. Extract searchblox.war, using for example the Java jar tool or a regular unzip command. On Windows, applications like WinZip work as well. Extract the files into a directory or sub-directory named searchblox, for example C:\xxxx\searchblox

2. Delete searchblox/WEB-INF/lib/commons-logging.jar

3. Move searchblox/WEB-INF/lib/log4j-*.jar into $JRUN/servers/lib/ (create this directory if it doesn't exist)

4. Logon to the JRun Management Console.

5. In the left pane of the Console, Select default | J2EE Components.

6. In the right pane of the Console, click on the Add button under Web Applications.

7. In the field Source File Path, enter the name of the directory where the files extracted from searchblox.war are present.

8. Click on the Deploy button.

9. Under "General Settings for searchblox", enter /searchblox in the Context Path field and click Apply.

10. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.4 Oracle Application Server

Using SearchBlox with Oracle 10g Release 2

1. Deploy searchblox.war by using the "Deploy WAR file" functionality in the Oracle Enterprise Manager. Set "Application Name" to searchblox and "Map to URL" to /searchblox.

2. Check if SearchBlox has been deployed to the directory %OracleAS_HOME%\j2ee\home\applications\searchblox directory

3. Edit web.xml located in %OracleAS_HOME%\j2ee\home\applications\searchblox\searchblox\WEB-INF so that the file looks like this (the changes are marked bold):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
   Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
   <web-app>
<!-- SearchBlox Servlets -->
<listener>
<listener-class>com.searchblox.index.ContextListener</listener-
   class>
</listener>
<servlet>
<servlet-name>LicenseServlet</servlet-name>
<servlet-class>com.searchblox.admin.LicenseServlet</servlet-class>
</servlet>
<servlet>
<servlet-name>UserServlet</servlet-name>
<servlet-class>com.searchblox.admin.UserServlet</servlet-class>
</servlet>
<servlet>
```

```
<servlet-name>LoginServlet</servlet-name>
<servlet-class>com.searchblox.admin.LoginServlet</servlet-class>
</servlet>
<servlet>
<servlet-name>TemplateServlet</servlet-name>
<servlet-class>com.searchblox.admin.TemplateServlet</servlet-
   class>
</servlet>
<servlet>
<servlet-name>CollectionServlet</servlet-name>
<servlet-class>com.searchblox.admin.CollectionServlet</servlet-
   class>
</servlet>
<servlet>
<servlet-name>SearchServlet</servlet-name>
<servlet-class>com.searchblox.search.SearchServlet</servlet-class>
</servlet>
<servlet>
<servlet-name>IndexerServlet</servlet-name>
<servlet-class>
com.searchblox.admin.IndexerServlet</servlet-class>
</servlet>
<servlet>
<servlet-name>ReportServlet</servlet-name>
<servlet-class>
com.searchblox.report.ReportServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>LicenseServlet</servlet-name>
<url-pattern>/servlet/LicenseServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>UserServlet</servlet-name>
<url-pattern>/servlet/UserServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>LoginServlet</servlet-name>
<url-pattern>/servlet/LoginServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>TemplateServlet</servlet-name>
<url-pattern>/servlet/TemplateServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>CollectionServlet</servlet-name>
<url-pattern>/servlet/CollectionServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>SearchServlet</servlet-name>
<url-pattern>/servlet/SearchServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>IndexerServlet</servlet-name>
<url-pattern>/servlet/IndexerServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>ReportServlet</servlet-name>
<url-pattern>/servlet/ReportServlet</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>search.jsp</welcome-file>
```

```
</welcome-file-list>
<error-page>
<exception-type>java.lang.Exception</exception-type>
<location>/admin/errorpage.jsp</location>
</error-page>
<security-role>
<role-name>Admin</role-name>
</security-role>
<security-constraint>
<web-resource-collection>
<web-resource-name>Forbidden</web-resource-name>
<url-pattern>/config.xml</url-pattern>
<url-pattern>/license.xml</url-pattern>
</web-resource-collection>
<auth-constraint>
<role-name>Admin</role-name>
</auth-constraint>
</security-constraint>
</web-app>
```

4. Edit jazn-data.xml located in %OracleAS_HOME%\j2ee\home\application-deployments\searchblox so that the file looks like this:

```
<?xml version=”1.0” encoding=”UTF-8” standalone=’yes’?>
<!DOCTYPE jazn-data PUBLIC “JAZN-XML Data” “http://xmlns.oracle.
   com/ias/dtds/jazn-data.dtd”>
<jazn-data>
<!-- JAZN Realm Data -->
<jazn-realm>
<realm>
<name>jazn.com</name>
<users>
<user>
<name>SearchBlox</name>
</user>
</users>
<roles>
<role>
<name>AdminRole</name>
<members>
<member>
<type>user</type>
<name>SearchBlox</name>
</member>
</members>
</role>
</roles>
</realm>
</jazn-realm>
<!-- JAZN Policy Data -->
<jazn-policy>
</jazn-policy>
<!-- Permission Class Data -->
<jazn-permission-classes>
</jazn-permission-classes>
<!-- Principal Class Data -->
<jazn-principal-classes>
</jazn-principal-classes>
<!-- Login Module Data -->
<jazn-loginconfig>
</jazn-loginconfig>
</jazn-data>
```

5. Edit orion-application.xml located in %OracleAS_HOME%\j2ee\home\application-deployments\searchblox so that the file looks like this:

   ```
   <?xml version=”1.0”?>
   <!DOCTYPE orion-application PUBLIC “-//ORACLE//DTD OC4J
      Application runtime 9.04//EN” “http://xmlns.oracle.com/ias/dtds/
      orion-application-9_04.dtd”>
   <orion-application deployment-version=”10.1.2.0.0” default-
      data-source=”jdbc/OracleDS” treat-zero-as-null=”true”>
   <web-module id=”searchblox” path=”searchblox.war” />
   <persistence path=”persistence” />
   <security-role-mapping name=”Admin”>
   <group name=”AdminRole” />
   </security-role-mapping>
   <!--principals path=”principals.xml” /-->
   <jazn provider=”XML’ location=”jazn-data.xml” />
   <log>
   <file path=”application.log” />
   </log>
   <namespace-access>
   <read-access>
   <namespace-resource root=””>
   <security-role-mapping>
   <group name=”jazn.com/administrators” />
   </security-role-mapping>
   </namespace-resource>
   </read-access>
   <write-access>
   <namespace-resource root=””>
   <security-role-mapping>
   <group name=”jazn.com/administrators” />
   </security-role-mapping>
   </namespace-resource>
   </write-access>
   </namespace-access>
   </orion-application>
   ```

6. Shutdown and restart OracleAS

7. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

Using SearchBlox with OC4J Standalone (9.0.4)

1. Create a new directory named searchblox in the $OC4J-HOME/j2ee/home/applications folder.

2. Extract searchblox.war into the /searchblox directory, using for example the Java jar tool or a regular unzip command. On Windows, applications like WinZip work as well.

3. Edit the file $OC4J-HOME/j2ee/home/config/application.xml and add the following descriptor within the <orion-application> element.

   ```
   <web-module id=”searchblox” path=”../../home/applications/
      searchblox”/>
   ```

4. Edit the file $OC4J-HOME/j2ee/home/config/http-web-site.xml and add the following descriptor within the <web-site> element.

   ```
   <web-app load-on-startup=”true” application=”default”
   name=”searchblox” root=”/searchblox” />
   ```

5. Place a copy of the file $OC4J-HOME/j2ee/home/applications/searchblox/WEB-INF/lib/dom4j.jar in the directory $OC4J-HOME/j2ee/home/applib

6. Edit the file $OC4J-HOME/j2ee/home/applications/searchblox/WEB-INF/web.xml so that the element <security-constraint> contains only the elements shown below:

   ```
   <security-constraint>
   <web-resource-collection>
   <web-resource-name>Forbidden</web-resource-name>
   <url-pattern>/config.xml</url-pattern>
   <url-pattern>/license.xml</url-pattern>
   <url-pattern>/index/*</url-pattern>
   <url-pattern>/docs/*</url-pattern>
   <url-pattern>/ext/*</url-pattern>
   <url-pattern>/publish/*</url-pattern>
   <url-pattern>/stylesheets/*</url-pattern>
   <url-pattern>/stopwords/*</url-pattern>
   <url-pattern>/logs/*</url-pattern><auth-constraint />
   </web-resource-collection>
   </security-constraint>
   ```

7. Start OC4J

8. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.5 Pramati

1. Start the Pramati Deploy Tool and connect to the Pramati Server.

2. Click on "Open" button in the toolbar. Select the file searchblox.war in the dialog box and click on "Open". The Deploy Tool will now display all the components in searchblox.war

3. Select the "Start the application on server restart" checkbox present at the bottom of the right panel.

4. Select "Web Properties" for searchblox.war in the Explore Panel in the Deploy tool (the Explore Panel is on the left side of the Deploy Tool). The Deploy Panel now displays the Context Root on the Right Panel.

5. Change the value for the Context Root from "searchblox.war" to "searchblox".

6. Click on "Save" button in the toolbar to save the archive settings.

7. Click on "Deploy" button in the toolbar. The Deployment Progress dialog box will indicate the status of the deploy operation. On successful deployment, the Task Panel at the bottom of the Deploy Tool will display "searchblox.war deployed".

8. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.6 Resin

1. Add the downloaded searchblox.war to the directory resin-3.0.x/webapps.

2. Start resin-3.0.x/bin/httpd.sh on Unix or resin-3.0.x/bin/httpd.exe on Win32

3. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.7 Sun Application Server

Note: This guide refers to Sun Application Server 8.

1. Extract searchblox.war, using for example the Java jar tool or a regular unzip command. On Windows, applications like WinZip work as well. Extract the files into a directory or sub-directory named searchblox, for example C:\xxxx\searchblox

2. Login to the Application Server Admin Console.

3. In the left pane of the Console, expand the Applications branch and click on the Web Applications folder. The Web Applications page will now be displayed in the right pane of the console.

4. Click on the Deploy... button on the Web Applications Page. The Deployment Page will now be displayed on the right pane.

5. Set the Upload File radio button to NO. In the “File or Directory” text box, enter the name of the directory where you have extracted the searchblox.war file.

6. Click on the Next button. The Deploy Web Module page will now be displayed on the right pane.

7. In the text boxes for Application Name and Context Root, enter searchblox. Select the Pre-compile JSP checkbox.

8. Click on the OK button. After a few seconds, the right pane will display the Web Applications page with the message Deployed Web Applications (1 items). The searchblox application will now be listed on the page.

9. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.8 Apache Tomcat

1. Drop searchblox.war into tomcat’s webapps folder and Restart tomcat.

2. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.9 BEA Weblogic Server

Note: SearchBlox must be deployed to the Weblogic Server using an exploded WAR directory. SearchBlox will NOT deploy correctly when deployed as a WAR archive

1. Extract searchblox.war, using for example the Java jar tool or a regular unzip command. On Windows, applications like WinZip work as well. Extract the files into a directory or sub-directory named searchblox, for example C:\xxxx\searchblox

2. Start the Weblogic Server Administration Console for the domain in which you will be working.

3. In the left pane of the Console, expand the Deployments folder, right-click Web Application Modules, and select Deploy a New Web Application Module. This initiates the Deployment Assistant.

4. Using the Deployment Assistant, locate the searchblox subdirectory which contains the exploded WAR files. Weblogic Server will deploy all components it finds in and below the specified directory.

5. When you have located the searchblox sub directory to configure, click Target Application.

6. If you have more than one server or cluster in your domain, select the one on which you want to deploy your new Web Application and click Continue. If you have just one server in your domain, go to the next step.

7. Enter searchblox as the name for the Web Application in the Name field. If you have more than one server or cluster in your domain, click whether you want to copy the file to each server.

8. Click Deploy. The Console displays the Deploy panel, which lists deployment status and deployment activities for the Web Application. After a few seconds, the status will change to "Success".

9. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.10 IBM Websphere Server

1. Start the IBM Websphere Application Server, if it is not running.

2. Open the Run the IBM Websphere Administrative Console, if it is not running.

3. Open the Applications > Install New Application page.

4. The Preparing Application Install page appears. Enter the local path to the searchblox.war that you have downloaded. Enter searchblox in the Context Root box and click Next.

5. Accept the default values on the Preparing Application Install page, if appropriate for your WebSphere configuration. Click Next.

6. In Step 1: Change the Application Name to searchblox. Click Next.

7. In Step 2: Map Virtual Hosts for Web Modules, select the virtual host or hosts in which to install the searchblox application. Click Next.

8. In Step 3: Map modules to Application Servers, if you have multiple application servers, select the application server in which to install the searchblox application. Click Next.

9. In Step 4: Summary, review the installation configuration and click Finish.

10. When the "Application searchblox installed successfully" message appears on the Installing page, click the Save to Master Configuration link, and click the Save button on the Save page to save your workspace.

11. Start the searchblox Application in the Enterprise Applications panel. Select the box next to searchblox, and click Start.

12. Go to http://yourhost/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.11 SearchBlox Server for Windows

1. Download and install the SearchBlox Server for Windows.

2. Start the SearchBlox Server by clicking on the “SearchBlox Server” DeskTop Icon or selecting “Start SearchBlox” from the Windows Start Menu.

3. Go to http://localhost:8080/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.12 SearchBlox Server for Mac OS X

1. Download and install the SearchBlox Server for Mac OS X.

2. Open the folder where SearchBlox has been installed.
   With the default settings in the installer, this folder will be /Applications/SearchBlox Server.

3. Start the SearchBlox Server by double-clicking on the file startSearchBlox.command. The SearchBlox Server will now start in a new Terminal window.

4. Go to http://localhost:8080/searchblox/admin/main.jsp in your browser and Login using username : admin / password : admin

2.2.13 SearchBlox on Linux/Unix

The SearchBlox RTF Document Parser uses the Java AWT/Swing toolkit. Since typical AWT/Swing toolkit implementations use X11 as the principal backend graphics engine, X11 services are often necessary. In the absence of the X11 services, the following error would occur when parsing RTF documents:

```
java.lang.InternalError: Can’t connect to X11 window server using ‘:
0.0’ as the value of the DISPLAY variable.
at sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)
```

To use the X11 services, you should start the Sun JDK 1.4 in the Headless mode. To do this, add a java argument of -Djava.awt.headless=true. For example, with Tomcat, add the following line in cataline.sh (or setclasspath.sh):

```
JAVA_OPTS=-Djava.awt.headless=true
```

Note that, it seems the presence of X11 runtime libraries is still needed, that is, an X11 environment must be installed into the system, but the X11 server itself need not be started. For example, for RedHat Linux, the minimum set of X11 packages needed is:

- XFree86
- XFree86-libs

3 - Administration

3.1 Starting SearchBlox

After successful deployment to your server, SearchBlox starts up automatically. You can verify if SearchBlox has started successfully by viewing the status. log in /searchblox/logs folder. If you use a server console, it will display the following message:-

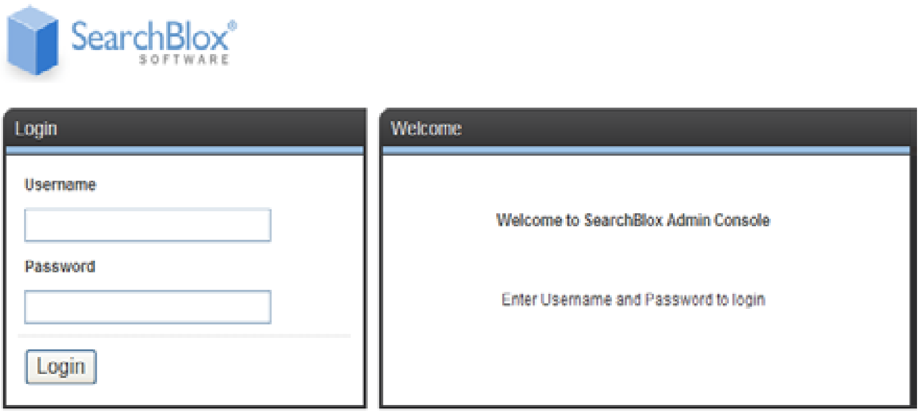
```
SearchBlox Version 4.0 Build 1
JVM Vendor: Sun Microsystems Inc.
JVM Version: 1.4.2_04
Server Information : WebLogic Server 8.1 SP3 Tue Jun 29 23:11:19 PDT
2004 404973 Version 2.3
OS Information : x86|Windows XP|5.1
Host IP Address : 192.168.0.2
SearchBlox Started....
```

The above message will be logged in the status.log file.

3.2 Accessing the Console

The Administration console for SearchBlox lets you do everything through a web based browser. You can access the console by typing in http://localhost:8080/searchblox/admin/main.jsp in your browser. When you access SearchBlox for the very first time, it will display the License Agreement. You can click on I Agree after reading through the License Agreement to get to the Log In page for the admin console. SearchBlox is pre-configured with the following username and password. Please login using the following credentials.

Username:	admin
Password:	admin



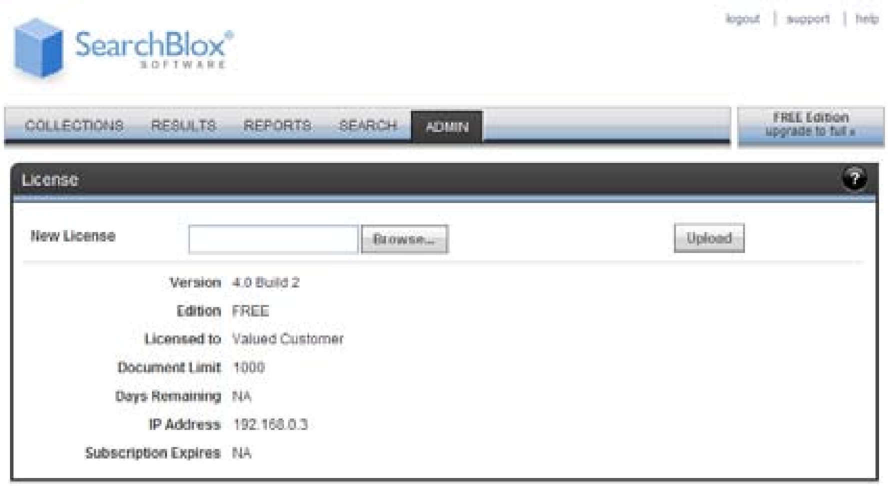
Welcome to SearchBlox Admin Console

Enter Username and Password to login

After you log-in you will be directed to the Admin Dashboard tab of the console. Click on the main tabs and each sub-tab under them to familiarize with the SearchBlox Admin console.

3.3 Verifying your Installation

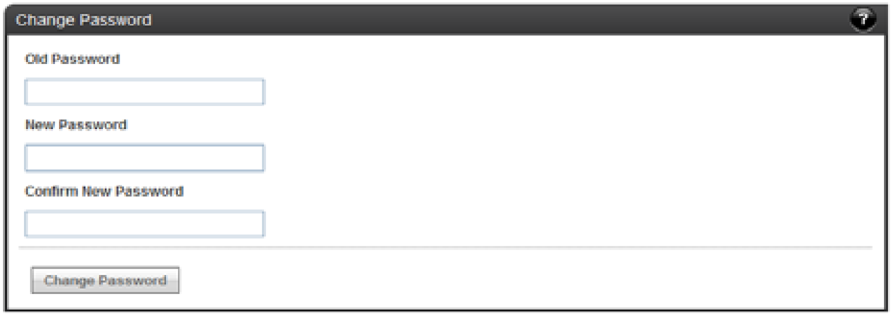
Click on the Admin tab to view your License information. If you have purchased SearchBlox, you will see the SearchBlox version and build number, Edition, Licensed Customer Name, Documents, IP Address and Support Subscription information here.



Version	4.0 Build 2
Edition	FREE
Licensed to	Valued Customer
Document Limit	1000
Days Remaining	NA
IP Address	192.168.0.3
Subscription Expires	NA

3.4 Changing your Admin Password

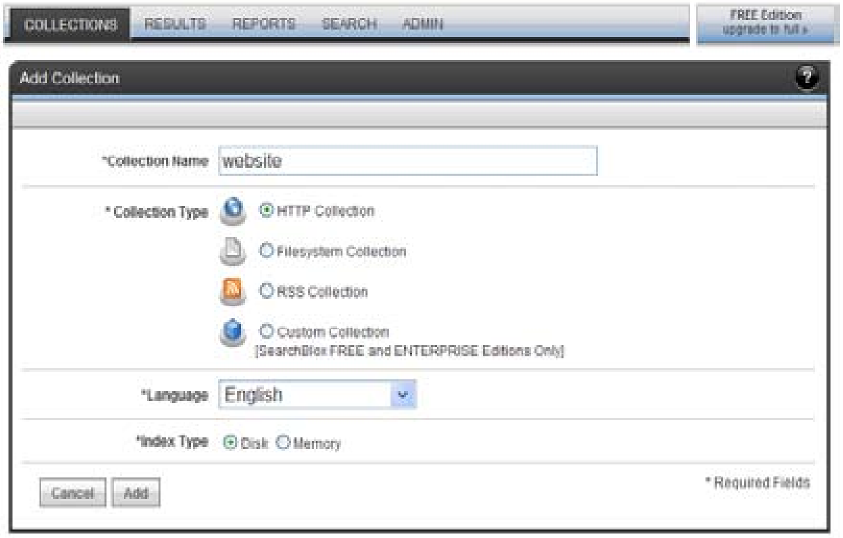
After you log-in the first time, please change your login password by clicking on the Admin tab. Your password is encrypted and stored securely.



3.5 Creating a HTTP Search Collection

SearchBlox includes a built-in web spider/crawler to index content from your intranet, portal or web site. This spider can index also index HTTPS content without any additional configuration. You can create a HTTP Search Collection by following the steps below.

1. After logging in to the Admin Console, click on the “Add Collection” button. The Add Collection screen will now be displayed.
2. Enter a unique name for your Collection (For example, ExternalWeb).
3. Click on HTTP radio button.
4. Choose the language of the web pages that you are about to index.
5. Select the Index Type for the collection (this determines where the working index will be placed)
6. Click Add to create the collection.



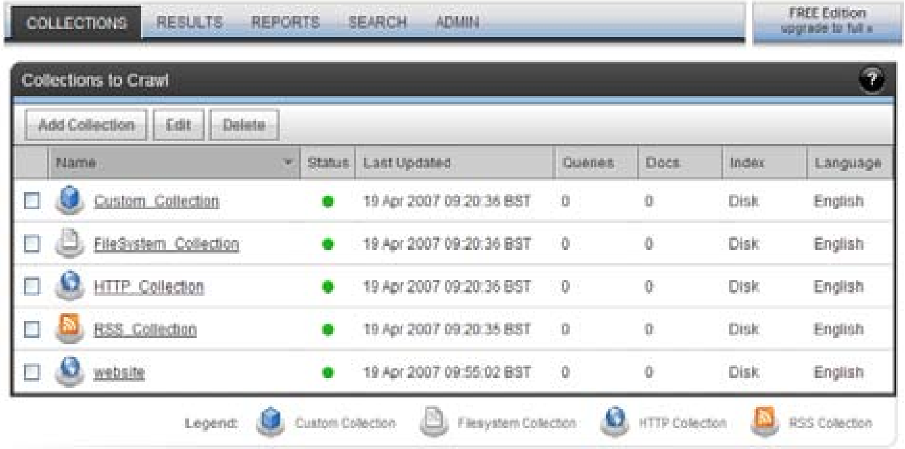
*Collection Name: website

* Collection Type:
○ HTTP Collection ● (selected)
○ Filesystem Collection
○ RSS Collection
○ Custom Collection
[SearchBlox FREE and ENTERPRISE Editions Only]

*Language: English

*Index Type: ● Disk ○ Memory

* Required Fields



	Name	Status	Last Updated	Queries	Docs	Index	Language
□	Custom_Collection	●	19 Apr 2007 09:20:36 BST	0	0	Disk	English
□	FileSystem_Collection	●	19 Apr 2007 09:20:36 BST	0	0	Disk	English
□	HTTP_Collection	●	19 Apr 2007 09:20:36 BST	0	0	Disk	English
□	RSS_Collection	●	19 Apr 2007 09:20:35 BST	0	0	Disk	English
□	website	●	19 Apr 2007 09:55:02 BST	0	0	Disk	English

Legend: Custom Collection, Filesystem Collection, HTTP Collection, RSS Collection

You should now see the collection you have created in the list of collections in the Collections tab.

After you have created the collection, follow the steps below to configure your collection.

3.5.1 Setting Paths

The Paths settings configure the Root URLs and the Filters for the collection. To access the Paths settings for the collection, click on the collection name in the   collections list.

3.5.1.1 Root URLs

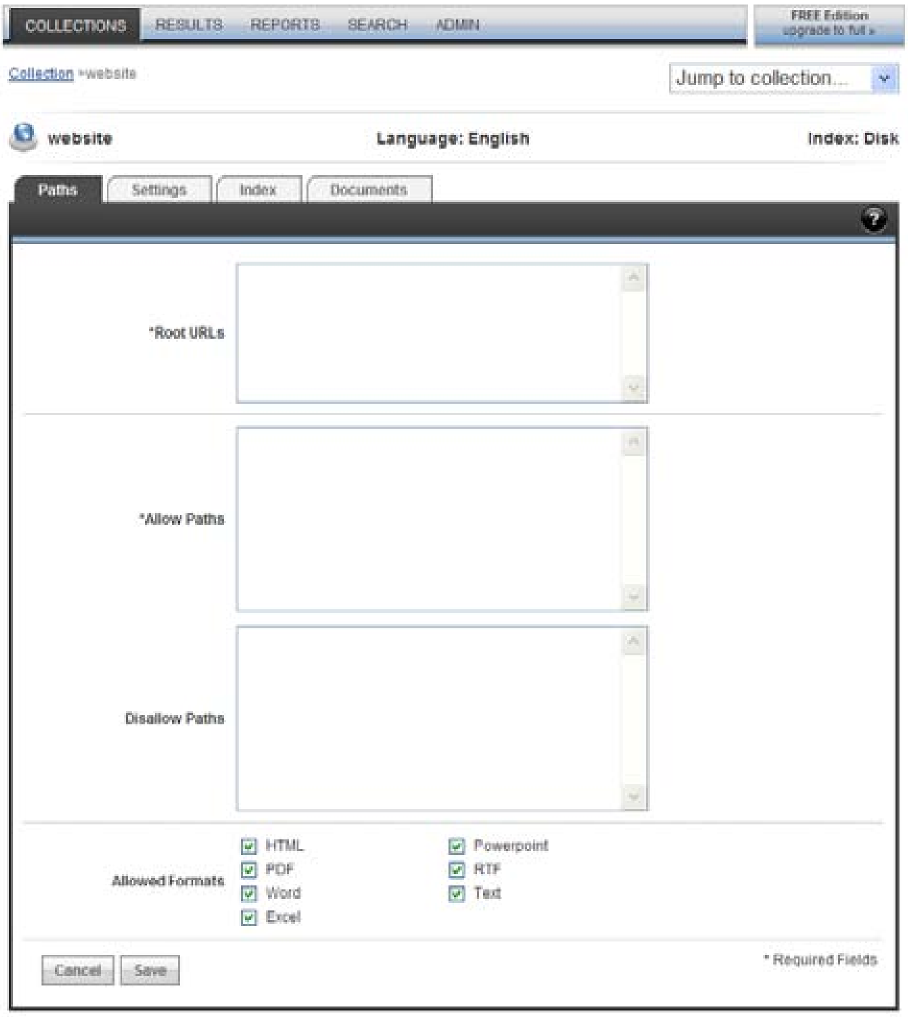
The Root URL is the starting URL for the spider. It requests this URL, indexes the content and follows links from this URL. Please make sure the Root URL you enter has regular html HREF links that the spider can follow.

1. In the Paths sub tab, enter at least one Root URL for the collection in the Root URLs text box. (For example, http://www.mywebsite.com/)
2. Click Save to save the values for the Root URLs.

3.5.1.2 Collection Filters

Filters let us configure the spider to include or exclude indexing URLs/ Documents. Allow and Disallow filters let us manage our collection by excluding unwanted URLs/documents.

Allow Paths	http://www.searchblox.com/.* (Informs the spider to stay only within the searchblox.com site.) .* (This lets the spider go anywhere it wants potentially indexing any site linked from the root URL)
Disallow Paths	.jsp /cgi-bin/.* /internal/.* (For http based collection)
Allow Formats	Select which formats are eligible to be part of your collection. SearchBlox currently supports the following file formats: HTML, PDF, WORD, EXCEL, POWERPOINT, RTF and TEXT.



*Root URLs

*Allow Paths

Disallow Paths

Allowed Formats
☑ HTML
☑ PDF
☑ Word
☑ Excel
☑ Powerpoint
☑ RTF
☑ Text

* Required Fields

3.5.2 Collection Settings

The Settings sub-tab holds tunable parameters for the spider. SearchBlox comes pre-configured with parameters when you create a new collection. We will walk you through various configuration settings and what they mean.

3.5.2.1 Keyword-in-Context Display

The Keyword-in-Context returns search results where the description displayed is from areas of content where the search term occurs. This feature comes pre-configured when you create a new collection. If want to disable this feature, Click on No radio button and Click Save at the bottom of the page.

3.5.2.2 HTML Parser Settings

This setting configures the HTML parser to read the description for a document from one of the following html tags:

Meta - <META name=”description” content=”some description here”>
H1 - <H1> some content here<H1>
H2 - <H2> some content here<H2>
H3 - <H3> some content here<H3>
H4 - <H4> some content here<H3>
H5 - <H5> some content here<H5>
H6 - <H6> some content here<H6>

If you disable the Keyword-in-Context feature, you can set description to be read from one of the above tags. If you don’t configure the description value when keyword-in-context is disabled, the description will be obtained by SearchBlox by reading the first 200 characters from the document.

3.5.2.3 Scanner Settings

These settings configure the HTTP spider/crawler with values to be used when requesting the content.

Maximum Document Age	(In days) specifies the maximum allowable age of a document in the collection.
Maximum Document Size	(In kilobytes) specifies the maximum allowable size of a document in the collection.
Maximum Spider Depth	Specifies the maximum depth the spider is allowed to proceed to index documents.
Spider Delay	Specifies the wait time in milliseconds for the spider between http requests to a web server.
User Agent	The name under which the spider requests documents from a web server.
Referrer	Is a URL value set in the request headers to specify where the user agent previously visited.
Ignore Robots	Value is set to Yes or No to tell the spider to obey robot rules or not.
Follow Redirects	Is set to Yes or No to instruct the spider to automatically follow redirects or not.

3.5.2.4 Boosting

You can boost search terms found in the Title, Description, Keywords or Body by setting a value greater than 1.

3.5.2.5 Stemming

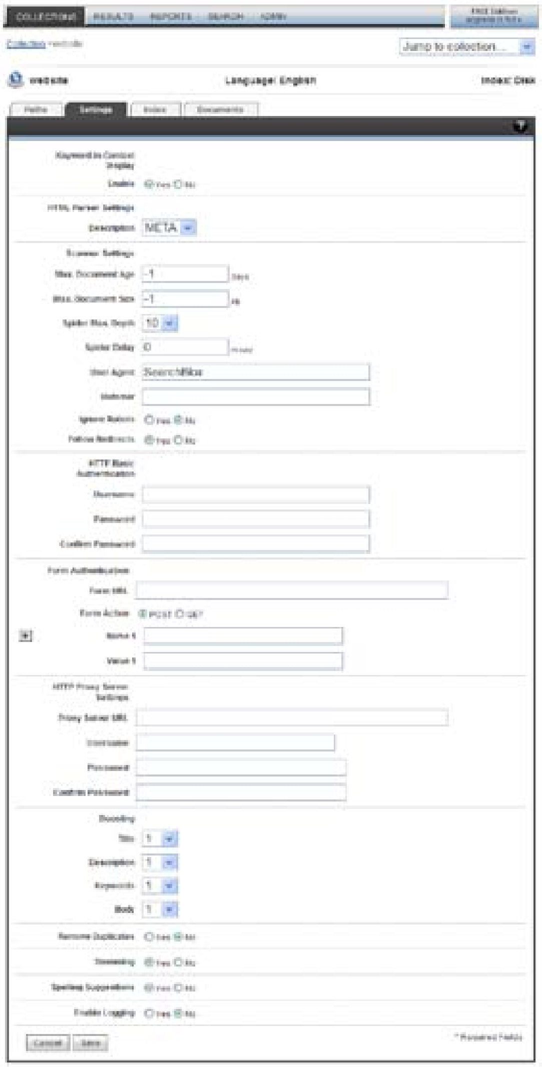
Stemming can be enabled or disabled for each individual collection. Stemming is enabled by default.

3.5.2.6 Spelling Suggestions

Spelling suggestions can be enabled or disabled for each individual collection. Spelling suggestions are enabled by default. When enabled, a spell index is created at the end of the index process. This spell index is located under /searchblox/spellindex and is named spellidxXXX.sdx where XXX is the collection ID. When enabled, the search results include the “Did you mean? “ functionality when the search query is misspelled.

3.5.2.7 Logging

You can set the Logging to DEBUG mode to view the indexer activity to troubleshoot issues with documents. Click on the Debug mode radio and click before you start indexing a collection.



website Language: English Index: Disk

Keyword in Context Display
Enable ● Yes ○ No

HTML Parser Settings
Description META

Scanner Settings
Max. Document Age -1 days
Max. Document Size -1
Spider Max. Depth 10
Spider Delay 0
User Agent SearchBlox
Webmaster
Ignore Robots ○ Yes ● No
Follow Redirects ● Yes ○ No

HTTP Basic Authentication
Username
Password
Confirm Password

Form Authentication
Form URL
Form Action ● POST ○ GET
Name 1
Value 1

HTTP Proxy Server Settings
Proxy Server URL
Username
Password
Confirm Password

Boosting
Title 1
Description 1
Keywords 1
Body 1

Remove Duplicates ○ Yes ● No
Stemming ● Yes ○ No
Spelling Suggestions ● Yes ○ No
Enable Logging ○ Yes ● No

* Required Fields

3.5.2.8 Indexing Content Secured by HTTP BASIC Authentication

When the spider requests a document, the spider presents these values (user/password) to the HTTP server in the “Authorization” MIME header. This authenticates the spider to the web server, which then lets the spider acccess URLs which are protected.

3.5.2.9 Indexing Content Secured by Form based Authentication

When access to documents is protected using form-based authentication, the spider can automatically login in and access the documents.

Form URL	is the ACTION URL of the authentication HTML form
Name/Value	is the set of name/ value pairs that are required. For example, username and password information for authentication are set here. Example: **Name** **Value** Webuser myself Password abc123 Login true
Form Action	specifies whether the form action is a POST or GET

3.5.2.10 Indexing Content using Proxy Servers

When access to HTTP content is through Proxy Servers, the Proxy Server settings are required are to enable the spider to successfully access and index content.

Proxy Server URL	is the URL to access the Proxy Server
Username / Password	When the Proxy Server requires authentication, set the username and the password.

3.5.3 Selective Indexing of Sections of HTML pages

With HTTP collections, there is often a requirement to exclude content from sections of a HTML page like headers, footers, navigation from being indexed. SearchBlox provides two ways to achieve this:

3.5.3.1 NoIndex Tags

SearchBlox supports the use of <noindex></noindex> to exclude content from being indexed as show below:

```
<noindex>
    Content to exclude
</noindex>
```

All information after the <noindex> tag and before the </noindex> tag will be excluded.

3.5.3.2 StopIndex Tag

SearchBlox supports the use of <!--stopindex--> <!--startindex--> to exclude content from being indexed as show below:

```
<!--stopindex-->
    Content to exclude
<!--startindex-->
```

All information before the <!--stopindex--> tag and after the <!--startindex--> tag will still be indexed.

3.6 Creating a File System Search Collection

SearchBlox includes a built-in file system spider/crawler to index content from your file system. You can create a File System Search Collection by following the steps below.

1. After logging in to the Admin Console, click on the “Add Collection” button. The Add Collection screen will now be displayed.
2. Enter a unique name for your Collection (For example, SalesDocs).
3. Click on File System radio button.
4. Choose the language of the web pages that you are about to index.
5. Select the Index Type for the collection (this determines where the working index will be placed)
6. Click Add to create the collection.

You should now see the collection you have created in the list of collections in the Collections tab.

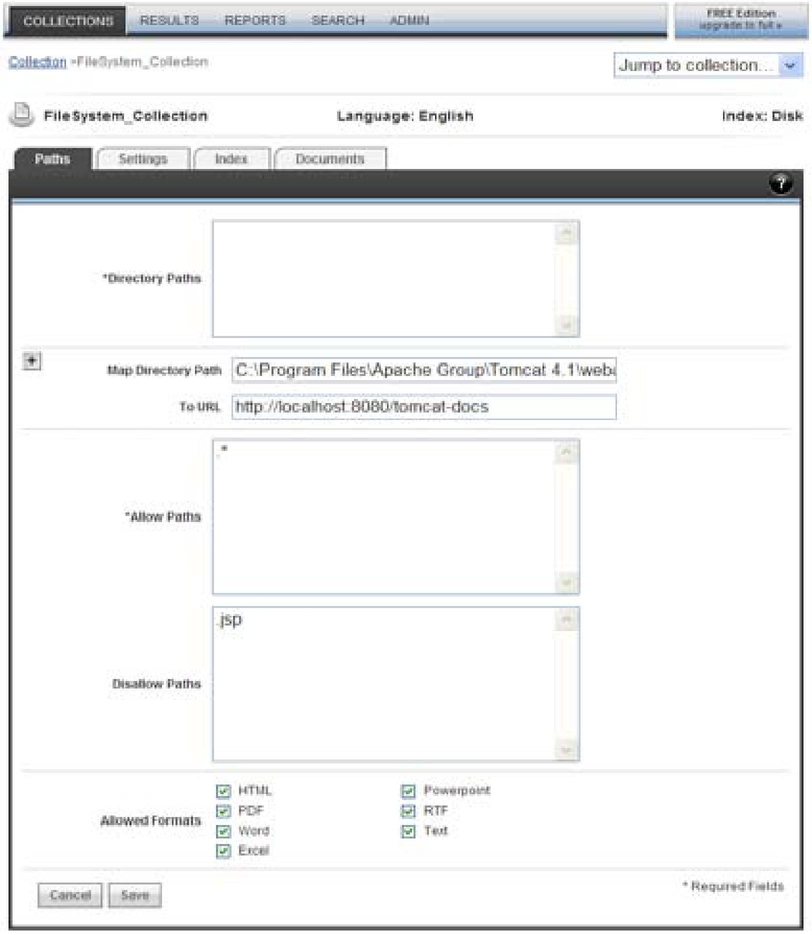
3.6.1 Setting Paths

The Paths settings configure the Directory Paths and the Filters for the collection. To access the Paths settings for the collection, click on the collection name in the collections list.

3.6.1.1 Directory Paths

The Directory Path is the starting path for the crawler. The crawler recursively indexes content within the folders.

1. In the Paths sub tab, enter at least one Directory path for the collection. (For example, c:\salesdocs or /var/web/html/salesdocs)
2. Click Save to save the values for the File System paths



FileSystem_Collection Language: English Index: Disk

*Directory Paths

Map Directory Path: C:\Program Files\Apache Group\Tomcat 4.1\web

To URL: http://localhost:8080/tomcat-docs

Allow Paths: .

Disallow Paths: .jsp

Allowed Formats: ☑ HTML ☑ PDF ☑ Word ☑ Excel ☑ Powerpoint ☑ RTF ☑ Text

* Required Fields

3.6.1.2 Collection Filters

Filters let us configure the spider to include or exclude indexing URLs/Documents. Allow and Disallow filters let us manage our collection by excluding unwanted URLs/documents.

Allow Paths	.* C:\\www\\html\\* (When creating a FileSystem based collection, specifying a allow filter is optional since the indexer is only going to look into sub-folders but if any symbolic links are placed, the spider will move to linked directories.)
Disallow Paths	C:\\www\\html\\noindex\\.* \\cgi-bin\\.* (For file system based collection)
Allow Formats	Select which formats are eligible to be part of your collection. SearchBlox currently supports the following file formats: HTML, PDF, WORD, EXCEL, POWERPOINT, RTF and TEXT.

3.6.1.3 Mapping file system paths to HTTP URLs

When you enter the directory path for a FileSystem Collection, you can optionally set an URL mapping for each directory. For example, C:\Program Files\Apache Group\Tomcat 4.1\webapps\tomcat-docs can be mapped to http://localhost:8080/tomcat-docs so that even though SearchBlox indexed the content from the file system, when a user clicks on the Search Result, the web document is served from the web server.

3.6.1.4 Configuring to serve content

When the Optional URL is not provided for the File System Collection, SearchBlox will automatically serve the files that appear in the search results.

3.6.2 Collection Settings

The Settings sub-tab holds tunable parameters for the FileSystem crawler and the indexer. SearchBlox comes pre-configured with parameters when you create a new collection. We will walk you through various configuration settings and what they mean. Select the collection for which you are about to customize the settings from the drop down list for making changes.

3.6.2.1 Keyword-in-Context Display

The Keyword-in-Context feature returns search results where the description displayed is from areas of content where the search term occurs. This feature is enabled when you create a new collection. To disable this feature, Click on the "No" radio button and Click Save at the bottom of the page.

3.6.2.2 Scanner Settings

Maximum Document Age	(In days) specifies the maximum allowable age of a document in the collection.
Maximum Document Size	(In kilobytes) specifies the maximum allowable size of a document in the collection.

3.6.2.3 Boosting

You can boost search terms found in the Title, Description, Keywords or Body by setting a value greater than 1.

3.6.2.4 Stemming

Stemming can be enabled or disabled for each individual collection. Stemming is enabled by default.

3.6.2.5 Spelling Suggestions

Spelling suggestions can be enabled or disabled for each individual collection. Spelling suggestions are enabled by default. When enabled, a spell index is created at the end of the index process. This spell index is located /searchblox/spellindex and is named spellidxXXX.sdx where XXX is the collection ID. When enabled, the search results include the "Did you mean? " functionality when the search query is misspelled.

3.6.2.6 Logging

You can set the Logging to Debug mode to view the indexer activity to troubleshoot issues with documents. Click on the Debug mode radio and click Save before you start indexing a collection.

3.7 Creating a Feed Search Collection

SearchBlox includes a built-in spider/crawler that can index RSS and Atom feeds. RSS and Atoms feeds are essentially XML files that provide information about recently changed content on a website or weblog. You can create a Feed Search Collection by following the steps below.

1. After logging in to the Admin Console, click on the "Add Collection" button. The Add Collection screen will now be displayed.
2. Enter a unique name for your Collection (For example, News).
3. Click on Feed radio button.
4. Choose the language of the web pages that you are about to index.
5. Select the Index Type for the collection (this determines where the working index will be placed)
6. Click Add to create the collection.

You should now see the collection you have created in the list of collections in the Collections tab.

3.7.1 Setting Paths

The Paths settings configure the Feed URLs and the Filters for the collection. To access the Paths settings for the collection, click on the collection name in the collections list.

3.7.1.1 Feed URLs

The Feed URL is the URL of the RSS/Atom feed page. The content of this URL is a XML file containing information about a list of URLs on the website. The SearchBlox Feed crawler indexes URLs using information in the RSS/Atom file. The Feed Crawler, however, does not follow links on the URLs contained in this XML file.

1. Enter at least one Feed URL for the collection.
   (For example, http://rss.cnn.com/rss/cnn_topstories.rss)
2. Click Save to save the values for the Feed URLs.

3.7.1.2 Collection Filters

Filters let us configure the spider to include or exclude indexing URLs/Documents. Allow and Disallow filters let us manage our collection by excluding unwanted URLs/documents.

Allow Paths	http://www.searchblox.com/.* (Informs the spider to stay only within the searchblox.com site.) .* (This lets the spider go anywhere it wants potentially indexing any site linked from the root URL)
Disallow Paths	.jsp /cgi-bin/.* /internal/.* (For http based collection)
Allow Formats	Select which formats are eligible to be part of your collection. SearchBlox currently supports the following file formats: HTML, PDF, WORD, EXCEL, POWERPOINT, RTF and TEXT.

3.7.2 Collection Settings

The Settings sub-tab holds tunable parameters for the spider. SearchBlox comes pre-configured with parameters when you create a new collection. We will walk you through various configuration settings and what they mean. Select the collection for which you are about to customize the settings from the drop down list for making changes.

3.7.2.1 Keyword-in-Context Display

The Keyword-in-Context feature returns search results where the description displayed is from areas of content where the search term occurs. This feature is enabled by default when you create a new collection. To disable this feature, Click on the "No" radio button and Click Save at the bottom of the page.

3.7.2.1 Scanner Settings

These settings configure the HTTP spider/crawler with values to be used when requesting the content.

Maximum Document Age	(In days) specifies the maximum allowable age of a document in the collection.
Maximum Document Size	(In kilobytes) specifies the maximum allowable size of a document in the collection.
Spider Delay	Specifies the wait time in milliseconds for the spider between http requests to a web server.
User Agent	The name under which the spider requests documents from a web server.
Referrer	Is a URL value set in the request headers to specify where the user agent previously visited.

3.7.2.3 Boosting

You can boost search terms found in the Title, Description, Keywords or Body by setting a value greater than 1.

3.7.2.4 Stemming

Stemming can be enabled or disabled for each individual collection. Stemming is enabled by default.

3.7.2.5 Spelling Suggestions

Spelling suggestions can be enabled or disabled for each individual collection. Spelling suggestions are enabled by default. When enabled, a spell index is created at the end of the index process. This spell index is located /searchblox/spellindex and is named spellidxXXX.sdx where XXX is the collection ID. When enabled, the search results include the "Did you mean? " functionality when the search query is misspelled.

3.7.2.6 Logging

You can set the Logging to DEBUG mode to view the indexer activity to troubleshoot issues with documents. Click on the Debug mode radio and click Save before you start indexing a collection.

3.7.2.7 Indexing Content Secured by HTTP BASIC Authentication

When the spider requests a document, the spider presents these values (user/password) to the HTTP server in the "Authorization" MIME header. This authenticates the spider to the webserver which lets the spider views URLs which are protected.

3.7.2.8 Indexing Content Secured by Form based Authentication

When access to documents is protected using form-based authentication, the spider can automatically login in and access the documents.

Form URL	is the ACTION URL of the authentication HTML form
Name/Value	is the set of name/ value pairs that are required. For example, username and password information for authentication are set here. Example: **Name** **Value** Webuser myself Password abc123 Login true
	specifies whether the form action is a POST or GET

3.7.2.9 Indexing Content using Proxy Servers

When access to HTTP content is through Proxy Servers, the Proxy Server settings are required are to enable the spider to successfully access and index content.

Proxy Server URL	is the URL to access the Proxy Server
Username / Password	When the Proxy Server requires authentication, set the username and the password.

3.8 Creating a Custom Collection

The Custom Collection enables indexing and searching of custom content. The indexing and deletion of documents in a Custom collection is done using the SearchBlox REST API. You can create a Custom Search Collection by following the steps below.

1. After logging in to the Admin Console, click on the "Add Collection" button. The Add Collection screen will now be displayed.
2. Enter a unique name for your Collection (For example, News).
3. Click on Custom radio button.
4. Choose the language of the web pages that you are about to index.
5. Select the Index Type for the collection (this determines where the working index will be placed)
6. Click Add to create the collection.

You should now see the collection you have created in the list of collections in the Collections tab.

3.8.1 Collection Settings

The Settings sub-tab holds tunable parameters for the parser. SearchBlox comes pre-configured with parameters when you create a new collection. We will walk you through various configuration settings and what they mean. Select the collection for which you are about to customize the settings from the drop down list for making changes.

3.8.1.1 Keyword-in-Context Display

The Keyword-in-Context feature returns search results where the description displayed is from areas of content where the search term occurs. This feature is enabled by default when you create a new collection. To disable this feature, Click on the "No" radio button and Click Save at the bottom of the page.

3.8.1.2 Scanner Settings

These settings configure the http scanner/spider with values to be used when requesting the content.

User Agent	The name under which the spider requests documents from a web server.
Referrer	Is a URL value set in the request headers to specify where the user agent previously visited.

3.8.1.3 Boosting

You can boost search terms found in the Title, Description, Keywords or Body by setting a value greater than 1.

3.8.1.4 Stemming

Stemming can be enabled or disabled for each individual collection. Stemming is enabled by default.

3.8.1.5 Spelling Suggestions

Spelling suggestions can be enabled or disabled for each individual collection. Spelling suggestions are enabled by default. When enabled, a spell index is created at the end of the index process. This spell index is located /searchblox/spellindex and is named spellidxXXX.sdx where XXX is the collection ID. When enabled, the search results include the "Did you mean? "functionality when the search query is misspelled.

3.8.1.6 Logging

You can set the Logging to DEBUG mode to view the indexer activity to troubleshoot issues with documents. Click on the Debug mode radio and click Save before you start indexing a collection.

3.8.1.7 Indexing Content Secured by HTTP BASIC Authentication

When the spider requests a document, the spider presents these values (user/password) to the HTTP server in the "Authorization" MIME header. This authenticates the spider to the webserver which lets the spider views URLs which are protected.

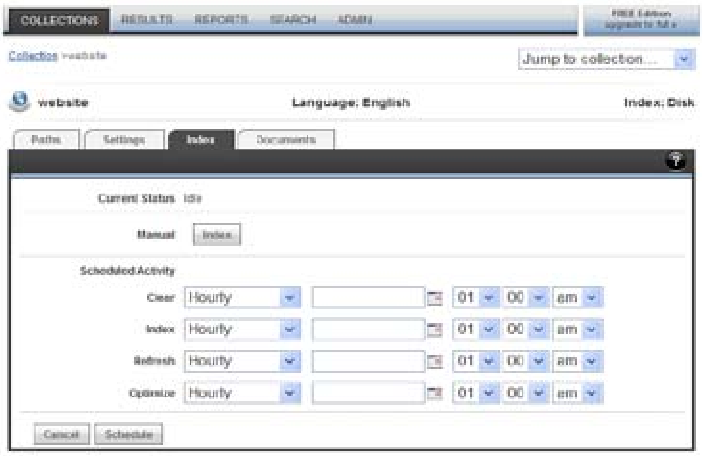
3.8.1.8 Indexing Content using Proxy Servers

When access to HTTP content is through Proxy Servers, the Proxy Server settings are required are to enable the spider to successfully access and index content.

Proxy Server URL	is the URL to access the Proxy Server
Username / Password	When the Proxy Server requires authentication, set the username and the password.

3.9 Indexing

Indexer Activity is controlled from Index sub tab for the collection. The current status of an indexer for a particular collection is indicated.



Collection > website

website Language: English Index: Disk

Current Status Idle

Manual

Scheduled Activity

Clear	Hourly		01	00	am
Index	Hourly		01	00	am
Refresh	Hourly		01	00	am
Optimize	Hourly		01	00	am

Index	Starts the indexer for the selected collection. Starts indexing from the root URLs or directory paths.
Clear	Clears the current index for the selected collection.
Refresh	Revisits URLs from the current index to make sure they are still valid and then continues to index newly discovered URLs.
Scheduled Activity	For each collection, you can set any of the following scheduled indexer activity: **Index** - Set the frequency and the start date/time for indexing a collection. **Refresh** - Set the frequency and the start date/time for refreshing a collection. **Clear** - Set the frequency and the start date/time for clearing a collection. **Optimize** - Set the frequency and the start date/time for optimizing a collection.

3.10 Collection Status

The Collections tab displays the name of the Collection, the language, collection type (HTTP or FileSystem), where the index is kept, the number of documents currently in the index, the number of queries that each collection has processesed and status of the indexer (Indexing or Ready).

3.11 Searching

You can search the collection by going to the Search Tab in the Admin Console or by pointing your browser to http://yourhost/searchblox/search.jsp.

3.11.1 Query Syntax

SearchBlox supports a wide range of searching options.

3.11.1.1 Wildcard Searches

SearchBlox supports single and multiple character wildcard searches. To perform a single character wildcard search, use the "?" symbol. To perform a multiple character wildcard search, use the "*" symbol.

The single character wildcard search looks for terms that match that with the single character replaced. For example, to search for "text" or "test" you can use the search:

```
te?t
```

Multiple character wildcard searches looks for 0 or more characters. For example, to search for test, tests or tester, you can use the search:

```
test*
```

You can also use the wildcard searches in the middle of a term.

```
te*t
```

*Note: You cannot use a * or ? symbol as the first character of a search.*

3.11.1.2 Fuzzy Searches

SearchBlox supports fuzzy searches. To do a fuzzy search, use the tilde, "~", symbol at the end of a Single word Term. For example to search for a term similar in spelling to "roam" use the fuzzy search:

```
roam~
```

This search will find terms like foam and roams

3.11.1.3 Proximity Searches

SearchBlox supports finding words are a within a specific distance away. To do a proximity search, use the tilde, "~", symbol at the end of a Phrase. For example to search for "searchblox" and "j2ee" within 10 words of each other in a document use the search:

```
"searchblox j2ee"~10
```

3.11.1.4 Boolean Operators

Boolean operators allow terms to be combined through logic operators. Lucene supports AND, "+", OR, NOT and "-" as Boolean operators (Note: Boolean operators must be ALL CAPS).

3.11.1.4.1 OR

The OR operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the OR operator is used. The OR operator links two terms and finds a matching document if either of the terms exist in a document. This is equivalent to union using sets. The symbol || can be used in place of the word OR.

To search for documents that contains either "searchblox server" or just "searchblox" use the query:

```
"searchblox server" searchblox
```

or

```
"searchblox server" OR searchblox
```

3.11.1.4.2 AND

The AND operator matches documents where both terms exist anywhere in the text of a single document. This is equivalent to an intersection using sets. The symbol && can be used in place of the word AND.

To search for documents that contain "searchblox server" and "j2ee" use the query:

```
"searchblox server" AND "j2ee"
```

3.11.1.4.3 +

The "+" or required operator requires that the term after the "+" symbol exist somewhere in a the field of a single document.

To search for documents that must contain "searchblox" and may contain "j2ee" use the query:

```
+searchblox j2ee
```

3.11.1.4.4 NOT

The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets. The symbol ! can be used in place of the word NOT. To search for documents that contain "searchblox server" but not "j2ee" use the query:

```
"searchblox server" NOT "j2ee"
```

Note: The NOT operator cannot be used with just one term. For example, the following search will return no results:

```
NOT "searchblox server"
```

3.11.1.4.5 -

The “-” or prohibit operator excludes documents that contain the term after the “-” symbol. To search for documents that contain “searchblox server” but not “j2ee” use the query:

> “searchblox server” -”j2ee”

3.11.1.4.6 Grouping

SearchBlox supports using parentheses to group clauses to form sub queries. This can be very useful if you want to control the Boolean logic for a query.

To search for either “searchblox” or “server” and “j2ee” use the query:

> (searchblox OR server) AND j2ee

This eliminates any confusion and makes sure you that website must exist and either term searchblox or server may exist.

3.11.1.4.7 Escaping Special Characters

SearchBlox supports escaping special characters that are part of the query syntax. The current list special characters are

> + - & || ! ( ) { } [ ] ^ “ ~ * ? : \

To escape these character use the \ before the character. For example to search for (1+1):2 use the query:

> \(1\+1\)\:2

3.11.1.5 Fielded Searches

SearchBlox supports fielded search. When performing a search you can either specify a field, or use the default field. The field names and default field is fixed and cannot be changed. You can search any field by typing the field name followed by a colon “:” and then the term you are looking for. Below is the list of fields available in SearchBlox:

Fields	Default	Range of Values	Description
content	Yes		Document content
keywords	No		Document keywords
description	No		Document description
title	No		Document title
url	No		Document URL
content-type	No	pdf - Adobe Acrobat PDF (.pdf) word - Word (.doc) excel - Excel (.xls) ppt - PowerPoint (.ppt) rtf - Rich Text Format (.rtf) txt - Text (.txt)	Document format

Fields	Default	Range of Values	Description
language	No	ar – Arabic bn – Bengali zhcn - Chinese (Simplified) zhtw - Chinese (Traditional) cs – Czech da – Danish nl – Dutch en – English es – Estonian fi – Finnish fr – French de – German el – Greek gu – Gujarati iw – Hebrew hi – Hindi hu – Hungarian it – Italian ja – Japanese kn – Kannada ko – Korean lv – Latvian lt – Lithuanian mal – Malayalam no – Norwegian pl – Polish pt – Portuguese ru – Russian ro – Romanian sk – Slovak sl – Slovenian es – Spanish sv – Swedish ta – Tamil te – Telugu th – Thai tr – Turkish	Document language

3.12 Scheduled Operations

For each collection, regular maintenance operations can be scheduled. These operations can be scheduled for a specific date and time and to repeat on varying frequencies.

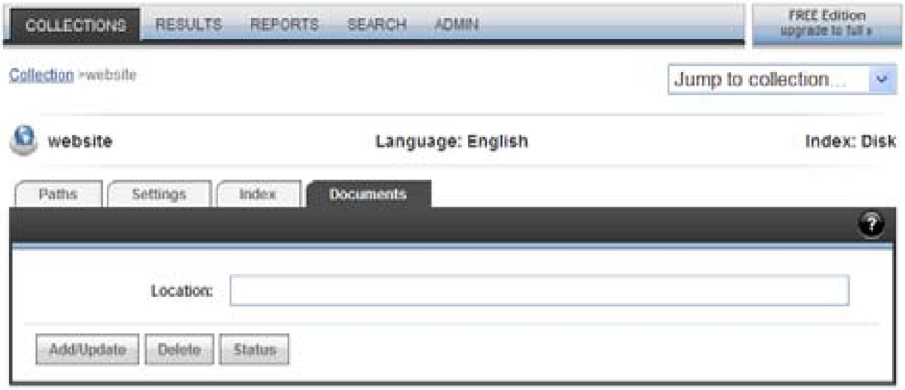
Here is the description of the available scheduled operations.

Index	Every document is indexed. If the document already exists in the index, it is deleted and re-indexed.
Refresh	Only documents that have changed since they were last added to the index are indexed. If new documents are found, they will be added to the index. If any document has been deleted since they were added to the index, they will be removed from the index.
Optimize	Optimizing makes the index optimized for fast searching. The optimize operation is automatically carried out at the end of the Index and Refresh operation. The optimize operation has to be manually done if individual documents have been manually added or deleted to the index using the SearchBlox Admin Console.
Clear	All the documents are deleted from the index

3.13 Adding/Updating/Deleting Individual Documents

Individual documents can be added/updated/deleted from a collection through the Documents sub-tab for the collection in the Admin Console.

Add	Add an individual document for indexing to the selected collection
Update	Update an individual document from the selected collection
Delete	Delete an individual document from the selected collection
Status	Get the status of a document from the selected collection. The Title, Description, Keywords, Size, URL, Last Modified Date, Index Date of a document can be seen using the Status function.

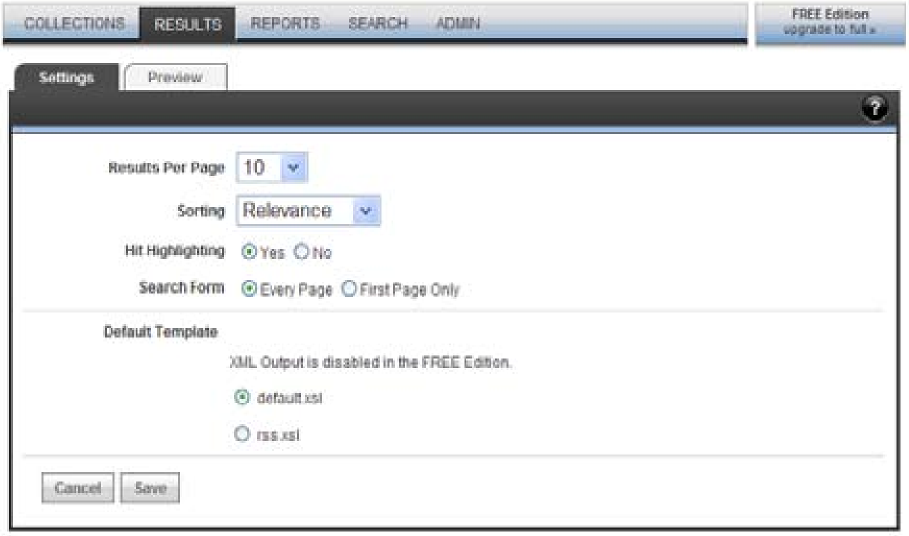


website Language: English Index: Disk

Location:

3.14 Changing Search Results Display through the Console

The search results settings can be configured using the Results tab on the Admin Console. They specify the **number of results** to display, the default **Sorting** method to be used, whether to provide **Hit Highlighting** and whether the **search form** should be displayed on every page. These settings are customizable per collection by overriding them through query parameters.



Results Per Page: 10

Sorting: Relevance

Hit Highlighting: ● Yes ○ No

Search Form: ● Every Page ○ First Page Only

Default Template

XML Output is disabled in the FREE Edition.

● default.xsl

○ rss.xsl

3.15 Logging

SearchBlox provides logging of all key activities that occur in the application. There are 3 logs which are located in the /logs directory under the searchblox base directory. All logs are rotated on a daily basis.

3.15.1 Status Log

All status messages and major error messages are logged in this log file. Shown below is the log output when SearchBlox starts up.

```
INFO  <21 Dec 2004 11:46:04,179> <status> <SearchBlox Version 2.0
  Build 30>
INFO  <21 Dec 2004 11:46:04,279> <status> <JVM Vendor: Sun
  Microsystems Inc.>
INFO  <21 Dec 2004 11:46:04,279> <status> <JVM Version: 1.4.2_05>
INFO  <21 Dec 2004 11:46:04,279> <status> <Server Information :
  Apache Tomcat/4.1.30 Version 2.3>
INFO  <21 Dec 2004 11:46:04,279> <status> <OS Information :
  x86|Windows XP|5.1>
INFO  <21 Dec 2004 11:46:04,279> <status> <Host IP Address :
  192.168.0.2>
INFO  <21 Dec 2004 11:46:04,960> <status> <ReportEngine
  Optimized>
INFO  <21 Dec 2004 11:46:12,882> <status> <SearchBlox Started....>
```

3.15.2 Index Log

All details of indexing operations are available in this log. This is the log to check to see whether a document has been indexed or skipped by the spiders. It also has the time taken to index each document.

```
INFO  <20 Dec 2004 10:57:31,379> <index> <SearchBlox Started....>
INFO  <20 Dec 2004 10:58:32,887> <index> <tomcat-docs> <Indexed:
  file:/C:/Program Files/Apache Group/Tomcat 4.1/webapps/tomcat-docs/
  BUILDING.txt> <Time:470 msecs>
INFO  <20 Dec 2004 10:58:33,899> <index> <tomcat-docs> <Indexed:
  file:/C:/Program Files/Apache Group/Tomcat 4.1/webapps/tomcat-docs/
  cgi-howto.html> <Time:221 msecs>
INFO  <20 Dec 2004 10:58:34,379> <index> <tomcat-docs> <Indexed:
  file:/C:/Program Files/Apache Group/Tomcat 4.1/webapps/tomcat-docs/
  class-loader-howto.html> <Time:350 msecs>
INFO  <20 Dec 2004 10:58:35,962> <index> <tomcat-docs> <Indexed:
  file:/C:/Program Files/Apache Group/Tomcat 4.1/webapps/tomcat-docs/
  cobwebsearch.ppt> <Time:1583 msecs>
INFO  <20 Dec 2004 10:58:41,820> <index> <tomcat-docs> <Indexed:
  file:/C:/Program Files/Apache Group/Tomcat 4.1/webapps/tomcat-docs/
  example1.pdf> <Time:5718 msecs>
INFO  <20 Dec 2004 10:59:02,019> <index> <tomcat-docs> <Skipped
  C:\Program Files\Apache Group\Tomcat 4.1\webapps\tomcat-docs\
  appdev\sample\build.xml>
INFO  <20 Dec 2004 10:58:41,970> <index> <tomcat-docs> <Indexed:
  file:/C:/Program Files/Apache Group/Tomcat 4.1/webapps/tomcat-docs/
  example2.pdf> <Time:150 msecs>
```

3.15.3 Query Log

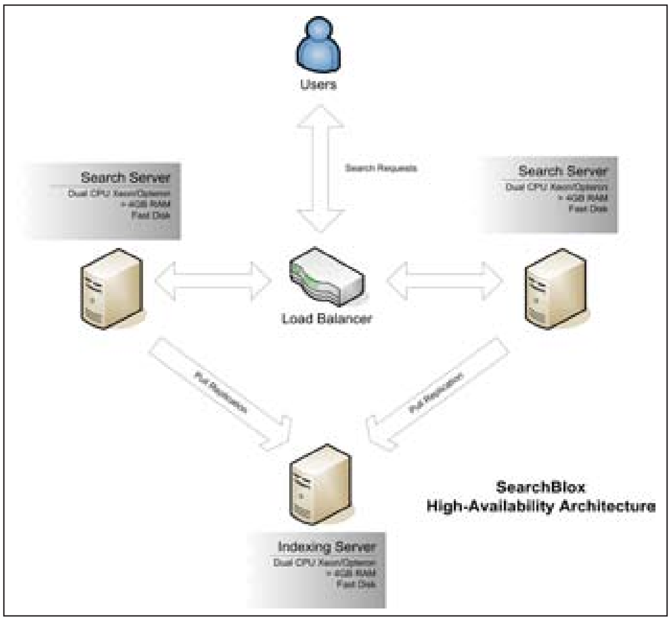
All search queries are logged in this file. This file can be used to generate customized search usage reports. It contains all available information regarding a search query and how it was processed.

```
INFO  <30 Dec 2004 10:57:41,765> <query> <SearchBlox Started....>
INFO  <30 Dec 2004 11:03:33,882> <query> <IP Address:127.0.0.1>
  <Collections:basic ja test > <Hits: 155> <Time: 80> <Query: java>
```

IP Address	IP Address of the search user
Collections	Name of the SearchBlox collections that were used for search
Hits	Number of hits or matches
Time	Time (in milliseconds) taken to return the search results
Query	The search query

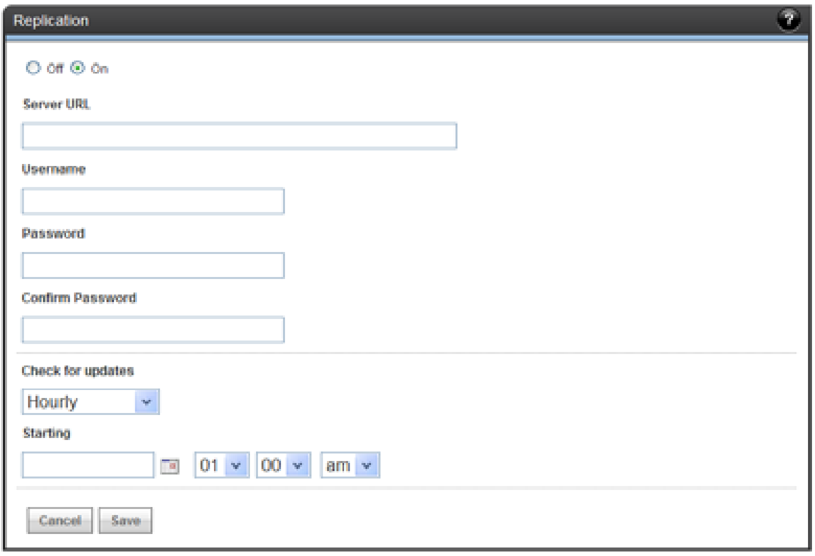
3.16 Replication

The Replication feature in SearchBlox (available in all ENTERPRISE Editions) keeps search indexes in multiple instances of SearchBlox across multiple physical machines synchronized. The SearchBlox instance that makes the indexes available for replication is called the Indexing Server and the one or more instances of SearchBlox that get the search indexes from the Indexing Server are called the Search Servers. The Indexing Server and the Search Servers use HTTP to communicate with each other. When changes are available on the Indexing Server, the Search Server downloads (Pull Replication) the changes, again over HTTP. Using SearchBlox Replication, a high-availability search architecture can be quickly and easily built as shown below:



3.16.1 Replication Settings

The Replication settings can be in the Admin tab in the SearchBlox Admin Console. If replication is disabled, click on the “On” radio button to display all the replication settings. *(Note: the Replication settings will be visible only in the SearchBlox ENTERPRISE Editions).*



Replication

○ Off ● On

Server URL

Username

Password

Confirm Password

Check for updates

Hourly

Starting

01 00 am

Server URL	The URL of the SearchBlox instance you are replicating. The Server URL must include the context root. Example: http://<hostname:port>/searchblox
Username	The Admin Console username of the SearchBlox instance that you are replicating
Password	The Admin Console password of the SearchBlox instance that you are replicating
Check for Updates	This specifies how often the server should be checked for index/file changes
Starting	Specifies the date/time for starting the replication process

Once the Replication settings are saved, the replication process will start at the specified time and keep the replicated server synchronized with the main server.

To disable replication, select the “Off” radio button in the Replication settings and click on the Save button.

3.16.2 Replication Log

When Replication is enabled, the details of the activities that occur during replication are logged to the replication.log file located in the /logs directory under the searchblox base directory.

4 - Customization

4.1 Add a search box to your web site to access SearchBlox

By adding a form to your web pages, you can let users search your documents/website indexed by SearchBlox.

```
<form name=”search” action=”http://www.searchblox.com/searchblox/
  servlet/SearchServlet” method=”POST”>
Search <input type=”Text” value=”” name=”query” size=”20” >
<input type=”image” name=”search” src=”http://www.searchblox.com/
  images/searchicon.jpg” alt=”Go” border=”0”>
</form>
```

Replace the form action parameter with your installation of SearchBlox like localhost:8080 instead of www.searchblox.com

4.2 Editing the default style sheet to add your company logo

To customize the look and feel of your search results page, edit the default.xsl file under stylesheets folder. You can replace the SearchBlox logo with your company logo, just find where it says “logo.jpg” in the xsl file and replace it with your own company logo file. You should place the company logo image file in the images folder under SearchBlox. To change other look and feel aspects of the page edit the HTML as required.

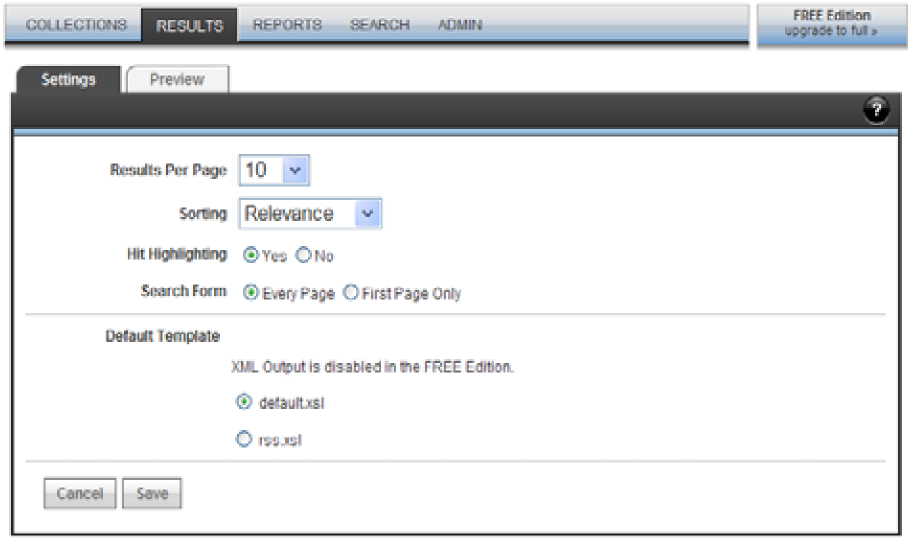
4.3 Changing display of search results using alternative style-sheets

You can specify the default xsl to be used in the Results tab of the Admin Console. To add a new template, place the stylesheet file in the /searchblox/stylesheets folder .

If you want to use a different XSL style sheet dynamically at request time, you can do that by passing the xsl parameter as part of the search request.

For example:

```
http://localhost/searchblox/SearchServlet?query=time&xsl=new.xsl
```



Results Per Page 10

Sorting Relevance

Hit Highlighting ● Yes ○ No

Search Form ● Every Page ○ First Page Only

Default Template

XML Output is disabled in the FREE Edition.

● default.xsl

○ rss.xsl

4.4 Using Search Parameters

SearchBlox uses the following search parameters when processing a search request. These parameters can be placed in your form fields for greater customization of the search results.

4.4.1 Basic Search Parameters

Parameter	Default Value	Range of Value	Description
query		Valid query syntax	Search query
fe	UTF-8	Valid web page encoding	Encoding of the search form
col	All available Collections	Existing collection ID	Collection ID
page	1		Search results page number
page size	10	1 to 1000	Number of results per page
xsl	default.xsl	Any XSL file that is present in the /stylesheets directory	Name of the XSL file to use
cname	All available collections	Any Collection Name	Name of collection
filter		Valid query syntax	Filter query
startdate		YYYYMMDDHHMMSS where YYYY – is the year (eg: 2007) MM – is the month (01 to 12) DD – is the day (01 to 31) HH – is hours (00 to 23) MM – is minutes (00 to 59) SS – is seconds (00 to 59)	Start date for date range search
enddate		YYYYMMDDHHMMSS where YYYY – is the year (eg: 2007) MM – is the month (01 to 12) DD – is the day (01 to 31) HH – is hours (00 to 23) MM – is minutes (00 to 59) SS – is seconds (00 to 59)	End date for date range search

4.4.2 Advanced Search Parameters

Parameter	Default Value	Range of Value	Description
st	Adv	Adv	Search Type
q_all		Valid query syntax	All the words
q_phr		Valid query syntax	Exact Phrase
q_low		Valid query syntax	Least one word
q_not		Valid query syntax	Without the words
contenttype		pdf - Adobe Acrobat PDF (.pdf) word - Word (.doc) excel - Excel (.xls) ppt - PowerPoint (.ppt) rtf - Rich Text Format (.rtf) txt - Text (.txt)	Document Format
language		ar – Arabic bn – Bengali zhcn - Chinese (Simplified) zhtw - Chinese (Traditional) cs – Czech da – Danish nl – Dutch en – English es – Estonian fi – Finnish fr – French de – German el – Greek gu – Gujarati iw – Hebrew hi – Hindi hu – Hungarian it – Italian ja – Japanese kn – Kannada ko – Korean lv – Latvian lt – Lithuanian mal – Malayalam no – Norwegian pl – Polish pt – Portuguese ru – Russian ro – Romanian sk – Slovak sl – Slovenian es – Spanish sv – Swedish ta – Tamil te – Telugu th – Thai tr – Turkish	Language

Parameter	Default Value	Range of Value	Description
enddate		YYYYMMDDHHMMSS where YYYY – is the year (eg: 2007) MM – is the month (01 to 12) DD – is the day (01 to 31) HH – is hours (00 to 23) MM – is minutes (00 to 59) SS – is seconds (00 to 59)	End date for date range search
oc		all - anywhere in the document title - in the title content - in the content keywords - in the keywords description - in the description URL - in the URL	Where the keywords occur
fe	UTF-8	Valid web page encoding	Encoding of the search form
col	All available	Existing collection ID	Collection ID
page	1		Search results page number
page size	10	1 to 1000	Number of results per page
xsl	default.xsl	Any XSL file that is present in the /stylesheets directory	Name of the XSL file to use
cname	All available collections	Any Collection Name	Name of Collection
filter		Valid query syntax	Filter query

4.5 XML Search Results

This feature is only available in the Paid Editions of SearchBlox. The xml data can be consumed by any client and transformed into any required format.
To output the search results in xml format, you can change the default template to XML in the Results > Templates section of the Admin console. Alternatively, you can also pass the parameter as part of your search request.

```
http://www.searchblox.com/searchblox/servlet/SearchServlet?
  query=time&xsl=xml
```

This query will output the search results as XML (shown below as a collapsed XML tree).

```
<?xml version="1.0" encoding="UTF-8" ?>
<searchdoc>
<results hits="42" time="0.00" query="searchblx"
suggest="searchblox" filter="" sort="relevance" start="1" end="10"
currentpage="1" lastpage="5" startdate="0" xsl="xml">
+<result no="1">
+<result no="2">
+<result no="3">
+<result no="4">
+<result no="5">
+<links url="../servlet/SearchServlet?query=searchblox&filter=&sort=
  relevance&col=1&startdate=0&xsl=xml&page=">
+<searchform url="../servlet/SearchServlet" query="searchblox">
</searchdoc>
```

The SearchBlox XML Search results consist of 4 types of information:

4.5.1 Search Details

All information regarding the processed search query is encapsulated in the <results> element tag. Below is the list of information available in the attributes of this element.

hits	Number of hits or matches for the query
time	Time taken to return the results of the search query in seconds
query	The search query
suggest	Spelling suggestions if available
filter	Preset filter used for the query
sort	Indicates how the search results are sorted. The possible values for this attribute are relevance, date and alpha.
start	The hit or match number of the first search result in this page
end	The hit or match number of the last search result in this page
current page	This indicates the current page number for this set of search results
last page	This indicates the last possible page number for this set of search results
start date	Indicates the start date for the date range search
end date	Indicates the end date for the date range search
page size	Number of search results per page

4.5.2 Result Details

All information about each search result is contained in the <result> element and its child elements.

```
<result no="1">
        <url>http://www.searchblox.com/milestones.html</url>
        <lastmodified>18 Nov 2004 10:52:24 EST</lastmodified>
        <indexdate>15 Dec 2004 08:18:58 EST</indexdate>
        <size>6348</size>
        <title><highlight>SearchBlox</highlight>- Milestones </title>
        <alpha>SearchBlox - Milestones</alpha>
        <keywords>search engine, java search engine, search application,
              search tool, tomcat search, weblogic search, java war file, search
        </keywords>
        <contenttype>HTML</contenttype>
        <context>FREE License up to 1000 documents. Try a demo search !
              <highlight>SearchBlox</highlight> > Contact > Milestones
        Request Information About Us Milestones Milestones Some key
        milestones in SearchBlox's
        </context>
        <description>FREE License up to 1000 documents. Try a demo search !
        <highlight>SearchBlox</highlight> > Contact > Mile...
        </description>
        <language>en</language>
        <score>78</score>
</result>
```

no	Hit number of the search result
url	Search result document URL
lastmodified	Last modified date of document
indexdate	Date the document was indexed
title	Title of the document
alpha	The text for the search result that is used for alphabetical sorting

keywords	The keywords contained in the document
contenttype	The format of the document
context	This is the fragment of the content where the search query appears. This is available only when the documents have been indexed with the Keyword-In-Context feature enabled.
description	The description contained in the document
language	Language setting for the document
score	The relevance score for the document for this query

4.5.3 Links

The <links> element contains pre-canned page URLs to simplify the process of paginating through the search results. The page URLs available here includes links to specific page numbers, the next and previous pages to the current page and the URLs for the alternative sorting options.

```
        <links url=”../servlet/SearchServlet?query=searchblox&filter=&
sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=0&xsl=xml
&page=”>
        <link page=”1” url=”../servlet/SearchServlet?query=searchblox&filter=
sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=0&xsl=
xml&page=1” />
        <link page=”2” url=”../servlet/SearchServlet?query=searchblox&filter=
sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=0&xsl=
xml&page=2” />
        <link page=”3” url=”../servlet/SearchServlet?query=searchblox&filter=
sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=0&xsl=
xml&page=3” />
        <link page=”4” url=”../servlet/SearchServlet?query=searchblox&filter=
sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=0&xsl=
xml&page=4” />
        <link page=”5” url=”../servlet/SearchServlet?query=searchblox&filter
=&sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=0&xsl=
xml&page=5” />
  <link page=”next” url=”../servlet/SearchServlet?query=searchblox&filter=
&sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=0&xsl=
xml&page=2” />
        <link page=”date” url=”../servlet/SearchServlet?query=searchblox&filt
er=&sort=date&col=3&col=5&col=1&col=2&col=4&startdate=0&page=
1&xsl=xml” />
        <link page=”alpha” url=”../servlet/SearchServlet?query=searchblox&fil
ter=&sort=alpha&col=3&col=5&col=1&col=2&col=4&startdate=0&page=
1&xsl=xml” />
        <link page=”relevance” url=”../servlet/SearchServlet?query=searchblo
x&filter=&sort=relevance&col=3&col=5&col=1&col=2&col=4&startdate=
0&page=1&xsl=xml” />
        </links>
```

4.5.4 Search Form

The <searchform> element provides all the information to display the search form as part of the search results. The information available here includes the list of selected collections used for the query.

```
<searchform url=”../servlet/SearchServlet” query=”searchblox”>
        <collections>
          <collection id=”3” name=”French” checked=”true” />
          <collection id=”5” name=”German” checked=”true” />
          <collection id=”1” name=”News” checked=”true” />
          <collection id=”2” name=”SearchBlox” checked=”true” />
          <collection id=”4” name=”Spanish” checked=”true” />
        </collections>
</searchform>
```

5 - SearchBlox REST-API

The SearchBlox REST-API enables the indexing and searching of custom content using simple HTTP POST and GET actions. The REST-API can add and delete documents from Custom Collections only.

Note: This feature is available in SearchBlox FREE and ENTERPRISE Editions only

5.1 Indexing Documents

5.1.1 Index URL

http://localhost:8080/searchblox/api/rest/add

5.1.2 Document Syntax

```
<?xml version="1.0" encoding="utf-8"?>
<searchblox licenseid="302D02144F115712204DD54596EDDCCEDD284878D3BBFA130215008BF2501E38561A439C9260D509DBF12952619177">
<document colname="Custom" location="http://www.searchblox.com/my.html">
        <url>http://www.searchblox.com/FEATURES/</url>
        <title boost="1">SearchBlox Product Features</title>
        <keywords boost="1"></keywords>
        <content boost="1">test</content>
        <description boost="1">SearchBlox Content Search Software</description>
        <lastmodified>07 May 2005 06:19:42 GMT</lastmodified>
        <size>44244</size>
        <alpha>SearchBlox Features</alpha>
        <contenttype>HTML</contenttype>
        <uid>urlid12345</uid>
        <category>Home/Product</category>
        <category>Home/Features</category>
</document>
</searchblox>
```

XML tags in italics are optional. If included, these fields will override the information from the document.

5.1.3 Document Description

XML Tag	Attribute	Value
searchblox	Licenseid	License value from license.xml file
document	colname	Name of the Custom collection
document	Location	Value of file location (http://.. or c:\documents\...)
url		url value to be returned with search results
uid		assigned unique identifier for a document (the url is used as uid when unassigned)
title		Title value of search result
title	boost	Boost value range from 1-10
keywords		Keyword value for document
keywords	boost	Boost value range from 1-10
content		content for document
content	boost	Boost value range from 1-10
description		description for search result
description	Boost	boost value range from 1-10
lastmodified		Date in format 07 May 2005 06:19:42 GMT
size		size value
alpha		text value for sorting
contenttype		type of content value (HTML, PDF etc)
category		value of category

5.2 Deleting a Document

5.2.1 Delete URL

http://localhost:8080/searchblox/api/rest/delete

5.2.2 Document Syntax

```
<?xml version=”1.0” encoding=”utf-8”?>
<searchblox licenseid=”302D02144F115712204DD54596EDDCCEDD284878D
      3BBFA130215008BF2501E38561A439C9260D509DBF12952619177”>
<document colname=”Custom_Collection” uid=”http://www.searchblox.
      com/features.html”>
</document>
</searchblox>
```

5.2.3 Document Description

XML Tag	Attribute	Value
searchblox	licenseid	License value from license.xml file
document	colname	Name of the Custom collection
document	uid	Value of unique identifier (file location http://..., file://c:/documents/.. or assigned uid)

5.3 Document Status

5.3.1 Status URL

http://localhost:8080/searchblox/api/rest/status

5.3.2 Document Syntax

```
<?xml version=”1.0” encoding=”utf-8”?>
<searchblox licenseid=”302D02144F115712204DD54596EDDCCEDD284878D
3BBFA130215008BF2501E38561A439C9260D509DBF12952619177”>
<document colname=”Custom_Collection” uid=”http://www.searchblox.
com/features.html”>
      </document>
</searchblox>
```

5.3.3 Document Description

XML Tag	Attribute	Value
searchblox	licenseid	License value from license.xml file
document	colname	Name of the Custom collection
document	uid	Value of unique identifier (file location http://..., file://c:/documents/.. or assigned uid)

5.4 Clear Collection

5.4.1 Clear URL

http://localhost:8080/searchblox/api/rest/clear

5.4.2 Document Syntax

```
<?xml version=”1.0” encoding=”utf-8”?>
<searchblox licenseid=”302D02144F115712204DD54596EDDCCEDD284878D
3BBFA130215008BF2501E38561A439C9260D509DBF12952619177”>
<document colname=”Custom_Collection”></document>
</searchblox>
```

5.4.3 Document Description

XML Tag	Attribute	Value
searchblox	licenseid	License value from license.xml file
document	colname	Name of the Custom collection

5.4 Optimize Collection

5.4.1 Optimize URL

http://localhost:8080/searchblox/api/rest/optimize

5.4.2 Document Syntax

```
<?xml version=”1.0” encoding=”utf-8”?>
<searchblox licenseid=”302D02144F115712204DD54596EDDCCEDD284878D
3BBFA130215008BF2501E38561A439C9260D509DBF12952619177”>
<document colname=”Custom_Collection”></document>
</searchblox>
```

5.4.3 Document Description

XML Tag	Attribute	Value
searchblox	licenseid	License value from license.xml file
document	colname	Name of the Custom collection

5.5 Response Codes

5.5.1 Document Syntax

```
<?xml version=”1.0” encoding=”utf-8”?>
<searchblox>
<statuscode>100</statuscode>
<statusdesc>Document Indexed</statusdesc>
</searchblox>
```

5.5.2 Status Code Description

100	Document Indexed
101	Error Indexing Document
200	Document Deleted
201	Document requested for deletion does not exist
301	Document Not Found
400	Collection Cleared
401	Error Clearing Collection
500	Invalid Collection Name
501	Invalid Request
502	Invalid Document Location
503	Specified collection is not a CUSTOM collection
504	Licensed Document Limit Exceeded
601	Invalid License ID
602	Custom documents not supported by this edition of SearchBlox
700	Collection Optimized
701	Error Optimizing Collection

5.6 SearchBlox Development Environment

The SearchBlox Development Environment is a simple browser-based tool designed to help developers use the SearchBlox REST-API with minimum effort. This tool is part of the SearchBlox deployment and can be assessed at http://yourhost/searchblox/sde/index.jsp

6 - SearchBlox Distributed Edition

The Distributed Edition adds a new dimension to the search capabilities of SearchBlox - the ability to distribute the handling of search requests to hundreds, even thousands of servers. Each of these servers runs an identical, standalone and special SearchBlox search application that contains all the components required to handle the search requests. The net result of this approach is that the search application can be deployed to any number of servers, giving you unlimited power to handle search requests.

Akamai EdgeComputing Platform is a J2EE Application Platform that leverages the Akamai's EdgePlatform, the world's largest distributed computing network, consisting of more than 14,000 servers in more than 1100 networks in over 70 countries.

Distributing SearchBlox Search on the Akamai EdgeComputing Platform is a 3-step process.

6.1 Index

The content is indexed by creating a HTTP collection or a FileSystem collection. When a FileSystem collection is used, the Optional URL should be used for mapping the Directory Path for the Collection. This will ensure that the document URL for any search result is an HTTP URL.

6.2 Publish

Once the documents have been indexed, the SearchBlox Distributable Search Application must be created or "published". The SearchBlox Distributable Search Application is a standalone and special SearchBlox search application that contains the search indexes for the indexed documents and other web components required to handle the search requests. Publishing creates a file named "searchblox.war" in a directory /publish under the searchblox base directory. The SearchBlox Distributable Search Application contains all indexes of all the collections that are setup in SearchBlox. The Publish operation cannot create a SearchBlox Distributable Search Application with the indexes of selected collections only.

To Publish, click on the "Publish" button in the Collections | Indexer sub-tab of the SearchBlox Admin Console. When the button is clicked, the "Publish" button will disappear from the screen. The "Publish" button will reappear on the screen once the publish operation is complete. The Publish functionality is disabled during the indexing operations of any individual collection.

6.3 Deploy

The "published" SearchBlox Distributable Search Application can now be deployed on the Akamai EdgeComputing Platform. To deploy, the deploy settings for the Akamai EdgeComputing Platform must first be provided. Note: Customers require a subscription to the Akamai EdgeComputing Platform to be able to deploy SearchBlox search on this platform. For information on the Akamai EdgeComputing Platform, visit www.akamai.com.

FTP Server	FTP server address (e.g. edgecomp.upload.akamai.com). This is the FTP server to which the SearchBlox Distributable Search Application will be uploaded to.
FTP Directory	name of directory where the file should be put (blank if not required)
FTP Username	ftp username
FTP Password	ftp password
CCU Username	CCU user name The Content Control Utility (CCU) is a Web interface available on the Akamai EdgeControl Management Center that allows you to specify the refreshing of specific cached objects, or to remove all objects specified by CP Codes.
CCU Password	CCU password
CCU Email	Email address for notification emails after deployment on Akamai EdgeComputing Platform
TEST URL	WAR URL for the test network (e.g. http://searchblox. edgecomputing.org/12293/webapps/SearchBloxTEST.war)
LIVE URL	WAR URL for the live network (e.g. http://searchblox.edgecomputing.org/12293/webapps/SearchBlox LIVE.war)

Once the above Deploy Settings have been entered, click on the Save button to save them.

6.3.1 Deploying to the TEST Network

To deploy the SearchBlox Distributable Search Application to the Akamai EdgeComputing Test Network, click on the Deploy [TEST] button. This process involves transferring the published searchblox.war file by FTP to the FTP Server and loading the application on the Akamai EdgeComputing Test network using the TEST URL. To check the progress of the deployment, view the status.log file located under /searchblox/logs directory. When the CCU Email address has been provided an email will be sent to the address on completion of the operation from the Akamai EdgeComputing Platform.

6.3.2 Deploying to the LIVE Network

To deploy the SearchBlox Distributable Search Application to the Akamai EdgeComputing LIVE Network, click on the Deploy [LIVE] button. This process involves transferring the published searchblox.war file by FTP to the FTP Server and loading the application on the Akamai EdgeComputing Test network using the LIVE URL. When the CCU Email address has been provided an email will be sent to the address on completion of the operation from the Akamai EdgeComputing Platform.

6.4 Scheduled Operations

To automate the process, set the schedulers for the “Publish” and “Deploy” operations. You can automate the deployment to the LIVE network only. The time settings for the “Deploy” operation can be set to 5 minutes after “Publish” setting.

7 - Upgrading

7.1 Upgrading to Newer Version of SearchBlox

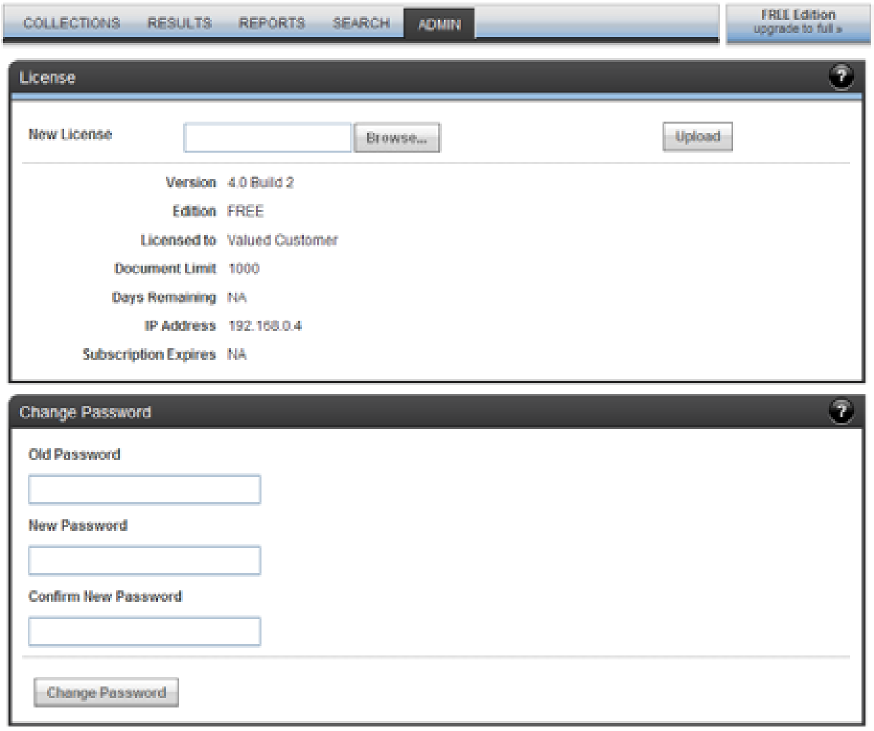
SearchBlox has the following user specific data.

Description	Location	Filename
User license file	/searchblox	license.xml
User configuration file	/searchblox	config.xml
Indexes	/searchblox/index	*.sdx
Reports	/searchblox/report	All files
Spell Indexes	/searchblox/spellindex	*.sdx
Stylesheets	/searchblox/stylesheets	*.xsl
Stopwords	/searchblox/stopwords	*.xml

7.2 Upgrading to a Higher Edition of SearchBlox

To upgrade to a higher SearchBlox Edition, purchase the required SearchBlox Edition and save the new license.xml on your system. Upgrade the license on your SearchBlox instance as follows:

1. After logging in to the Admin Console, click on the Admin tab.
2. Click on the Browse button and the use the file dialog to select the new license.xml file.
3. Click on the Upload button to upgrade the license.



Version 4.0 Build 2
Edition FREE
Licensed to Valued Customer
Document Limit 1000
Days Remaining NA
IP Address 192.168.0.4
Subscription Expires NA