

Extending Alerts

for Watson Explorer: Advanced Edition

Note: It is assumed that the reader is familiar with the
Watson Explorer: Analytics Edition (aka. WCA)

Users Guide

Date: August 29, 2014

Author: Todd Leyba (tleyba@us.ibm.com)

ABSTRACT

In May of 2014 Watson Content Analytics (version 3.5) was released with a new set of features - one of which included alerts. Alerts allow a user to be notified (via eMail or other means) when a certain condition has been met in the indexing of new or existing data. The alerting subsystem is quite robust allowing the creator to choose from a wide range of notification options, conditions, rules, thresholds and scheduling options. But with flexibility in options comes complexity and partly because of this it was decided to only allow the WCA administrator to create the alert. An end user can request an alert at any time during their use of the Content Miner or Enterprise Search application but not actually create the alert itself. Their request contains the current query at the point the request is being made along with any instructions from the user detailing their desired conditions for the alert. It is then up to the WCA Administrator to vet the request and actually create the alert or contact the requestor for further clarification.

The extension described in this paper allows the end user of WCA applications to actually create the alerts themselves - albeit with less complexity and options.

Disclaimer: While this asset increases the convenience of alert creation for the end user it can, if used improperly, cause unintended consequences for the WCA server and administrator. This alerting extension is not to be used by large numbers of Content Miner users (> 50) or should an individual user create a large number of alerts (> 100). By adhering to these recommended constraints your WCA administrator should experience little impact to the system while at the same time removing the need to cater to alert requests. If you plan to exceed these constraints then it is recommended that the administrator monitor the server performance carefully so as to determine the actual thresholds that are acceptable to your environment. Also it is the responsibility of the end user to maintain their own alerts and to remove alerts that are no longer needed.

With the above disclaimer understood the benefits to the end user are obvious and when used properly can leave the user with more control and faster gratification with alert notification. Let's look more closely at what the end user can actually see and do.

User Generated Alerts

The creation of alerts by an end user with this extension is a simplified process (as compared to that of the administrator). A few assumptions aid in this simplification process. First, it is assumed that the end user wants to be notified via email when the current conditions within your application are met or exceeded. Your current query represents those conditions to be met. For example, suppose you were searching WCA for any articles about "Machine Learning" technology. Your search may or may not have returned any results but you would like to be notified by email in the future when new articles arrive on the same topic.

A new tab (view) can be added to your application that allows the end user to manage their own alerts as shown below.

Managing Your Alerts

Alerts allow you to be notified when certain conditions arise in this collection of documents. Below are the alert conditions that represent your state of analysis and discovery as defined by the current query. Simply give your alert a name and provide the email address to which you want the notification sent. Note that you can optionally provide a description, modify the query itself or change how many results are required to trigger the alert. When ready press the "Add Alert" button.

*Name	Description	*eMail Address	*Current Query	*Min Results
Machine Learning	Alert me when for new articles about machine learning	tleyba@us.ibm.com	machine learning	1

List of Your Active Alerts:

Name	Description	Publisher	Alert Query	Min Results	Action
MachineLearning	Alert me when for new articles about machine learning	[esadmin] Mail Publisher	machine learning	1	<input type="button" value="delete"/>

The view is divided into two parts. The top portion allows you to create a new alert – the bottom portion lists all of your previously created and active alerts. To create an alert you simply fill out the required entries (indicated with a red asterisk) in the top row and then press the "Add Alert" button. The name for your alert must be unique. The email address is typically your own email address but can that of another person you wish to receive the alert notification. The query box will already be filled in with the current query. You can modify this query if need be. The minimum results number is the threshold of search results to equal or exceed in response to the alert query.

As you add alerts they will be placed at the top of your list of active alerts. You can remove an alert at any time by pressing its corresponding delete button.

The arrival of your alerts depends not only on the search results meeting the minimum threshold specified for the alert but also on the frequency with which the collection is updated. Consult with your administrator to determine how often the collection is updated.

Alert Extension Components

There are three major components that comprise the alert extensions to WCA. They are:

1. **Managing Alerts Custom Widget** – This is a custom view for your application that the user interacts with to create and manage their alerts (previously shown)
2. **Alert Extension Servlet** - The servlet that receives and processes requests from the user/view to add/delete alerts)
3. **Alert Extension Run Command** – This is a command line utility that when invoked will execute all user generated alerts. Which alerts actually result in notifications is dependent on their query and threshold settings. This command line utility is normally scheduled to run on a reoccurring basis by the host operating system.

Below is the architecture diagram for the components and where they fit in the Watson Content Analytics architecture.

Installing the Alert Extensions

Installing the Alert Extensions involves installing the three components previously mentioned and configuring those collections that are to be associated with the alert extensions. The three components are available in the WCAAlertExtensionsArchive.zip file accompanying this publication. Unzip the contents of the WCAAlertExtensionsArchive (.zip) file into a directory of your choice. Be sure to read the README.txt file for any last minute updates.

Prerequisites

The following prerequisites are required in order for the Alert Extensions to operate properly.

- **WCA Installed with WebSphere 8.0 or greater**

WebSphere is required to enable the Alert Extension Servlet to be deployed within the same web container as the WCA UI components. If the Alert Extension Servlet were deployed in another web container or application server then your browser would prevent the communication between the Alert Custom View and associated Servlet with a security violation. This is to protect against [Client Side Cross Domain Security Attacks](#). The embedded version of WebSphere (Liberty) does not have an administration console allowing you to deploy the Alert Extension Servlet into Liberty.

- **Application level security enabled in WebSphere**

It is also necessary to enable application level security within WebSphere so that the alerts can be associated with discrete user IDs. From the WAS admin console under Global Security ensure that the “Enable Application Security” box is checked (shown right).

If you are using the local operating system as the user registry then be sure to configure the primary administrative user id (shown right).

The image shows two screenshots from the WebSphere Security Configuration Wizard. The top screenshot displays the 'Administrative security' and 'Application security' sections. In the 'Application security' section, the checkbox for 'Enable application security' is checked and highlighted with a red box. The bottom screenshot shows the 'User account repository' section. The 'Current realm definition' is set to 'Local operating system'. In the 'Available realm definitions' section, 'Local operating system' is selected in the dropdown menu and highlighted with a red box, with a 'Configure...' button next to it.

Step 1.0 - Deploying the Alert Custom View

Deploying the Alert Custom View follows the same procedures as outlined in the Watson Content Analytics 3.5 Information Center on [“Creating and deploying a plug-in to add custom widgets for user applications”](#). They are repeated here in brief for your convenience.

Step 1.1) Change the hostname in the **AlertExtensionsSearchPane.js** in the **icaplugin** directory of the unzipped install component archive to your hostname. Use the hostname found in your WCA Search or Analytics application URL. Below are the lines that need to change in the **AlertExtensionsSearchPane.js** file highlighted in red:

```
hostname : "localhost",  
hostport : "80",
```

Step 1.2) Add the plug-in files to the ES_NODE_ROOT/master_config/searchapp/icaplugin directory.

Copy the **icaplugin** directory in the unzipped install component archive to the icaplugin directory in your **\$ES_NODE_ROOT\$\master_config\searchapp** directory (respond Yes when asked to replace existing files/directories). For example:

```
xcopy icaplugin\*.*  
C:\Program Files\IBM\ICA3.5\esadmin\master_config\searchapp\icaplugin
```

Step 1.3) Register the plugin.

- a. Back up and edit the appropriate widgets.json file for the type of application to which you want to add the custom widget:
 - o To register a custom widget for a search application, edit the ES_NODE_ROOT/master_config/searchserver/repo/search/*Application_Name*/widgets.json file.
 - o To register a custom widget for an analytics application, edit the ES_NODE_ROOT/master_config/searchserver/repo/analytics/*Application_Name*/widgets.json file.

Application_Name is the application ID, such as default, social, or advanced. You can determine the ID by viewing the list of applications in the application customizer.

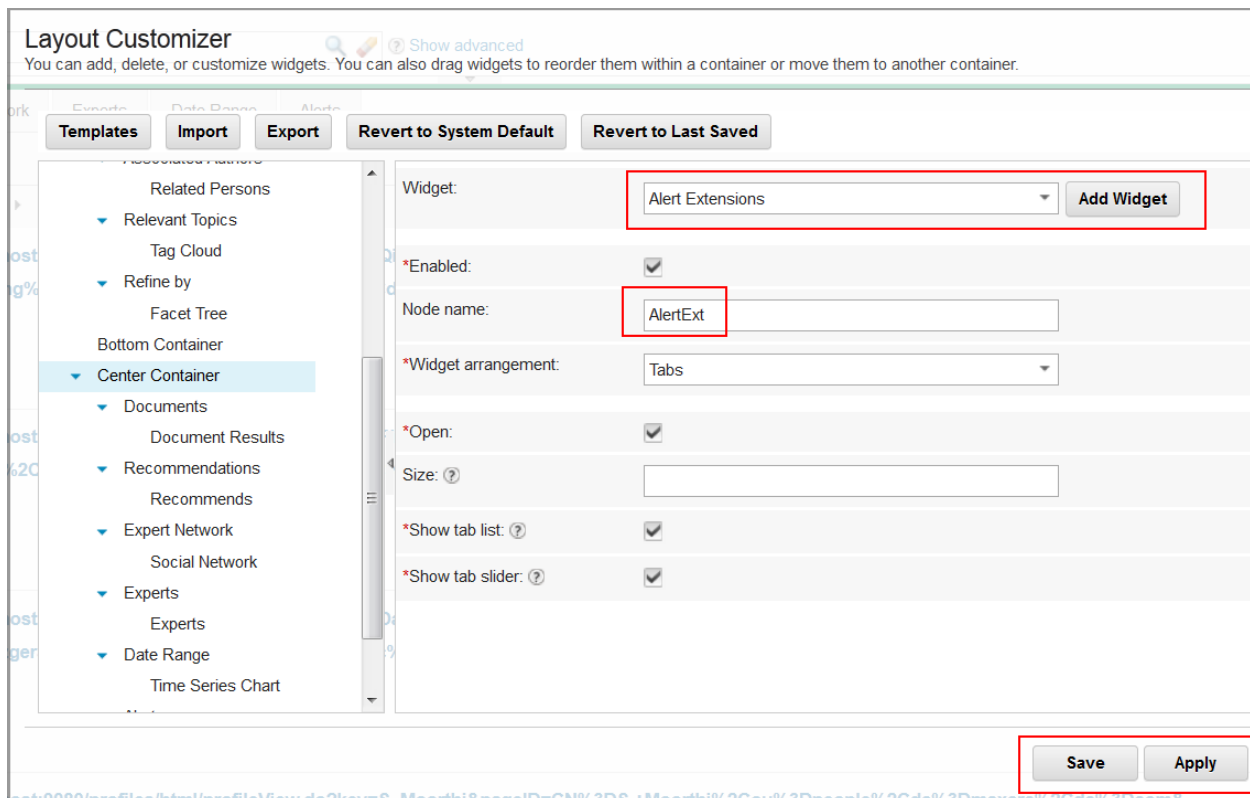
b. Add an entry for the widget in the following format:

```
    "AlertExtensionsSearchPane" : {  
      "available" : true,  
      "label" : "Alert Extensions" ,  
      "widgetName" : "icapplugin/AlertExtensionsSearchPane" ,  
      "properties": [{  
        "value":"test",  
        "name":"defaultQuery",  
        "editable":true,  
        "sync":false,  
        "type":"TextBox",  
        "label":"Default Query",  
        "widgetOptions":{},  
        "required":false}  
      ]  
    }
```

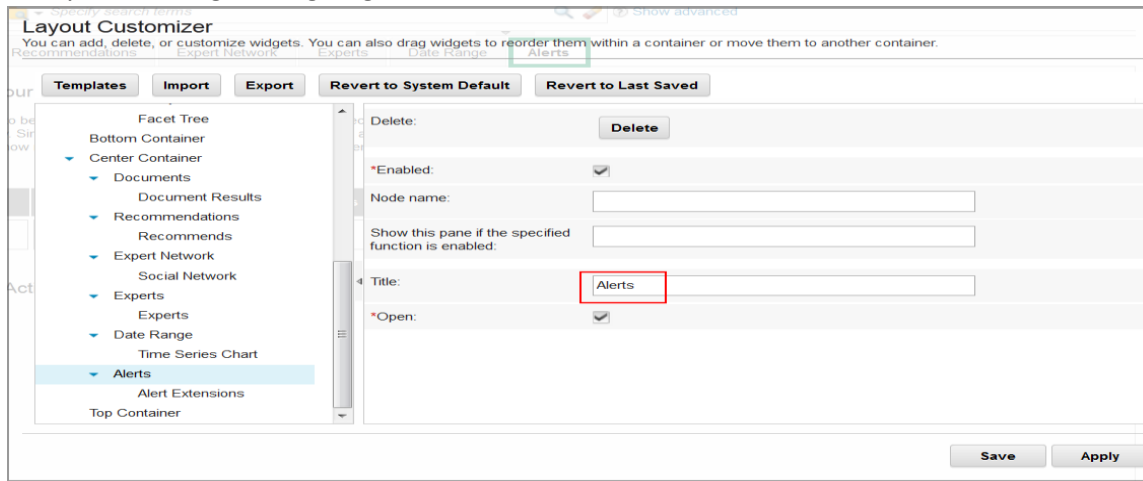
Be sure to stop and restart the system

Step 1.4) Add the Alert Widget to your application layout

From your application (type search or analytics) select “Customize Layout” from the menu bar pull down under your login ID. Highlight the layout section of the application that you would like to add the alert widget (“Center Container” is selected below). Select “Alert Extensions” as the type of widget and press the “Add Widget” button. Then press the “Apply” button to see how it looks in your application. If satisfied then press the “Save” button.



Note that you can change the name of the widget title (from New Pane to Alerts) by highlighting the newly added widget and giving it a new Title as shown below:

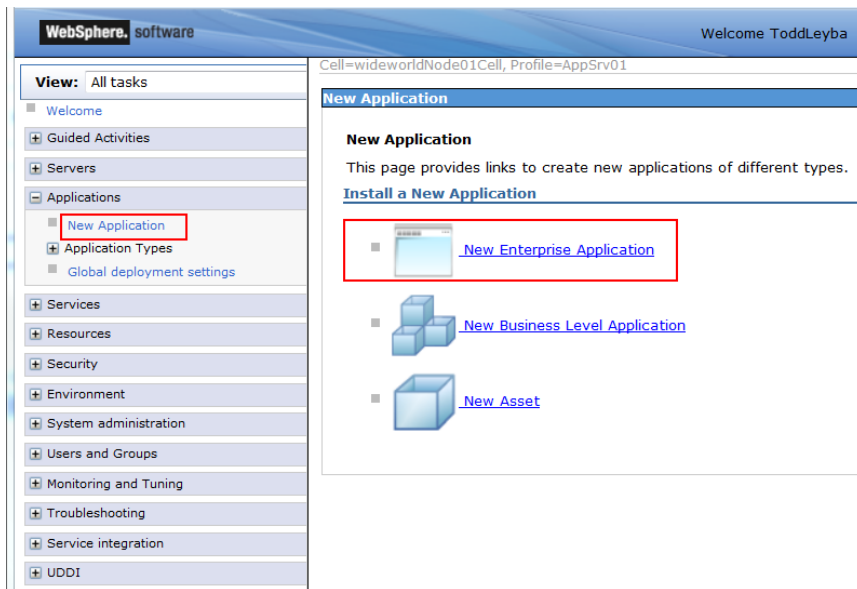


Your Alert Extension Widget should now be accessible from within your application. You can now move on to Deploying the Alert Extension Servlet

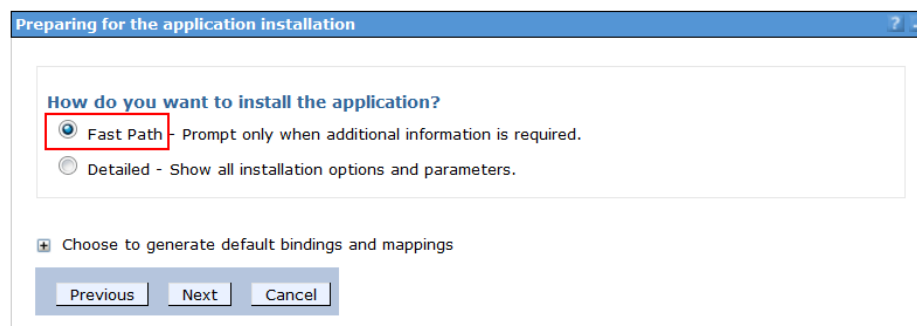
Step 2.0 - Deploying the Alert Extension Servlet

The Alert Extension Servlet is the web application that processes your alert requests made from the UI Alert Extension Widget. It must be deployed into the same WebSphere web container as the other web parts of WCA. Those WCA web apps (.ear files) are installed under a separate Web Application Server named ESSearchServer.

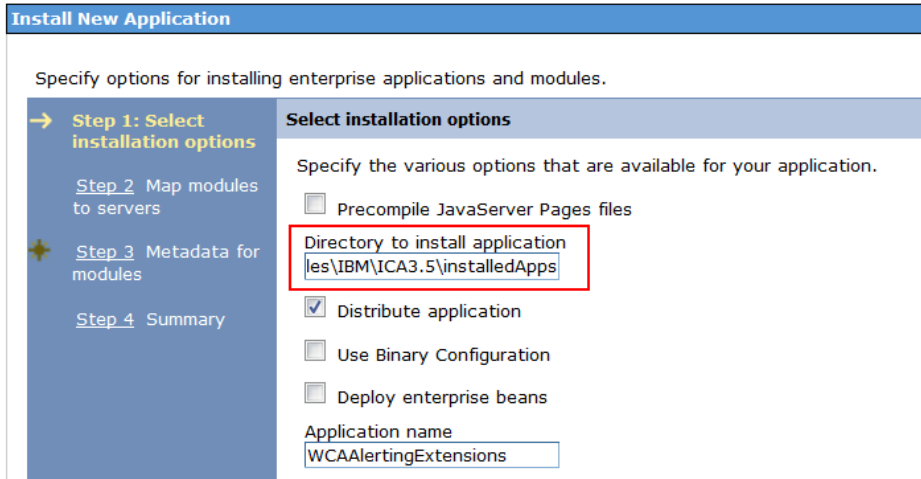
To start the deployment process log into your WebSphere Administration console and select Applications->New Application->New Enterprise Application



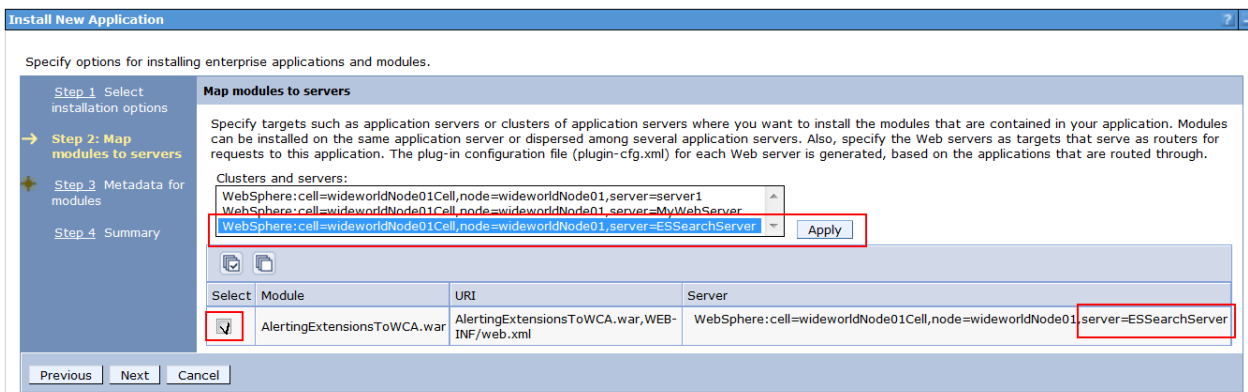
Then browse for and select the “WCAAlertingExtensions.ear” from the unzipped Alert Extensions Archive provided with this publication (and then press “Next”). Then select the “Fast Path” as shown below and then press “Next”.



For the directory to install the application enter the installApps directory in your WCA install directory as shown below, and then press “Next”.



On the next screen the Alerting Extensions module with a check mark and then highlight the ESSearchServer application server in the list above, and then press “Apply”. The application for the AlertExtensions should be changed to ESSearchServer (to the right). Press “Next” when changed.



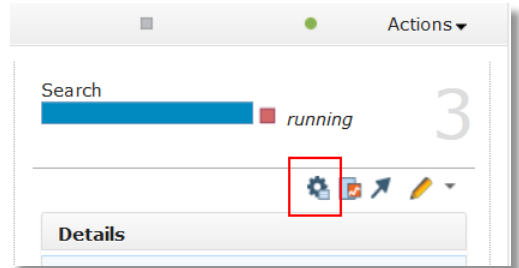
Press “Next” on the next screen (nothing to change) and on the Summary screen press “Finished”. Once the WCAAlertingExtensions.ear is successfully deployed be sure to press the “Save” button to save the changes to the master configuration of WebSphere. You are now ready to move to the last step of installation.

Restart the ESSearchServer for the deployment to be available.

Step 3.0 - Create a Template Alert Extensions Publisher

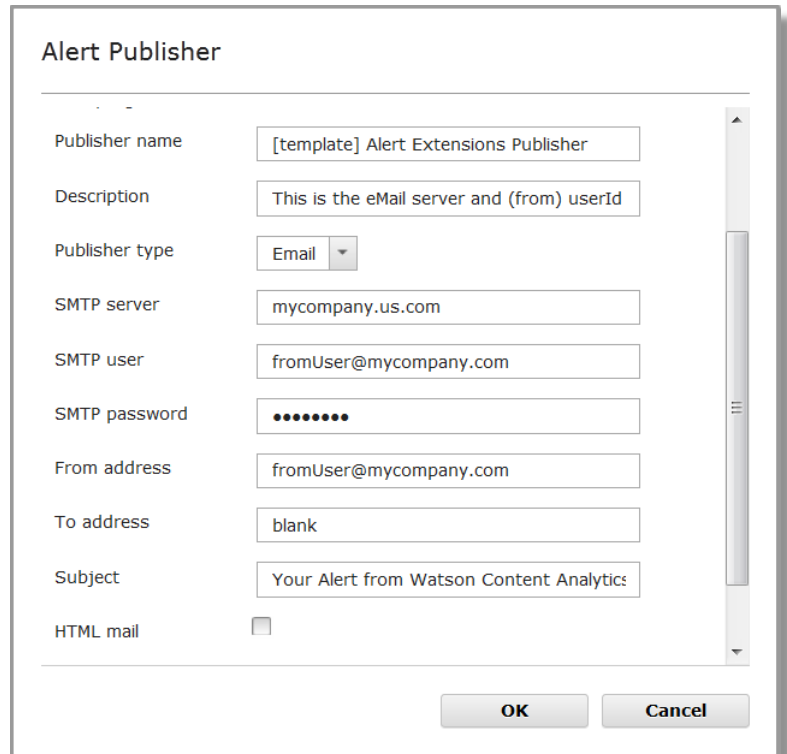
For each collection that you would like to use the Alert Extensions you need to create an email publisher template for the Alert Extensions. An email publisher template will be used by the Alert Extensions to create a single email publisher for each unique user creating and alert (regardless of how many alerts they create). The template is used to fill out the SMTP server name to be used along with the one userId and password that will be used to send the email alert when triggered.

From the administrator's console, open the collection and press the Alert icon in the Search Server portion of the collection as shown circled in red to the right:



Click on the "Create Publisher" button and enter the following information shown below:

The publisher name must be the string "[template] Alert Extensions Publisher". The alert extensions will look for the template using this name. The description can be anything. But the publisher type must be set to "Email". Provide your company's SMTP server name and the userId and Password to be used to send email notifications from that SMTP server. Notice that the SMTP userId and "From Address" are the same. The "To Address" can be left blank and will be filled in with the userId of each user creating an alert. The subject of the email can be anything you like but will be the same for all users requesting an alert from that collection.

A screenshot of a dialog box titled 'Alert Publisher'. It contains several input fields and a checkbox. The fields are: 'Publisher name' with the value '[template] Alert Extensions Publisher'; 'Description' with the value 'This is the eMail server and (from) userId'; 'Publisher type' with a dropdown menu set to 'Email'; 'SMTP server' with the value 'mycompany.us.com'; 'SMTP user' with the value 'fromUser@mycompany.com'; 'SMTP password' with a masked password '.....'; 'From address' with the value 'fromUser@mycompany.com'; 'To address' with the value 'blank'; 'Subject' with the value 'Your Alert from Watson Content Analytics'; and 'HTML mail' with an unchecked checkbox. At the bottom right, there are 'OK' and 'Cancel' buttons.

Press "Ok" when finished.

Step 4.0 - Schedule the Run Alert Extensions Command Utility

The *Run Alert Extensions Command Utility* is a command line program that when executed will run all alerts for a given collection. Whether or not an actual alert is triggered and a notification sent is dependent on the alert's query being satisfied for the current state of the collection.

You are to use your host operating system's scheduling service to schedule the execution of the run alert utility. For example, on Linux and AIX systems you can use the cron command and for Microsoft Windows you can use the Windows Task Scheduler Service.

In either case you should schedule the alert utility to run after the collection is expected to be built. Scheduling the alert utility to run more frequently than index builds would be a waste of resources and executions since nothing has changed in the index. It is also recommended that the alert utility be scheduled to run during off hours (perhaps during the early morning hours) so as not to compete and interfere with end user activity managing their own alerts during working hours.

The Run Alert Extensions Command Utility takes a single parameter that points to the location of a config.properties file whose contents should contain the following:

```
LoggingOn=false
LoggingFileName=C:\\temp\\RunWCAAlertExtensions.log
WCA_User=esadmin
WCA_Password=esadminpassword
WCA_Host=localhost
WCA_Port=8390
WCA_ColId=TESTCOL
WCA_DateFld=date
```

Note that a properties file should be created for each collection that uses the alert extensions and should be scheduled separately with the run alert utility. The WCA_ColId identifies the collection to run the alerts for.

The WCA_DateFld identifies the collection's index date field that is to be used by the alert extensions to determine which documents are new to the collection since the last run date of the run utility. The identified date field should contain the last crawl date and time of the document.

By default WCA crawlers set the reserved index "date" field to the date and time of the crawl. If your collection has mapped another field to the date field (e.g., date document was published) then you need to create another date field in the collection populated with the last crawl date and specify that date field name in the WCA_DateFld property. An alternative approach would be let the crawlers populate the default "date" field as normal, and create a different "published date" field for the document.