

Technical White Paper

Alphablox Alerts

ALPHABLOX ALERTS:

An In-Depth Technical View of Alphablox Alerts

TABLE OF CONTENTS

PURPOSE	3
INTRODUCTION	4
ALPHABLOX ALERTS	5
THE ALERTING PROCESS.....	8
DESIGNING AN ALERTING APPLICATION	12
DEVELOPING AN ALERTING APPLICATION.....	12
DESIGNING THE ALERT TEMPLATE	17
SUMMARY	19

Purpose

This white paper provides a technical overview of Alphablox Alerts and includes an introduction to the process of developing custom alerting applications. It is intended for IT application developers wishing to gain a clearer understanding of the Alphablox Alerts architecture. This paper should also give them an understanding of what an alerting application is and how it is developed using Alphablox Alerts.

Alphablox Alerts can be used as a standalone application to provide critical monitoring and notification capabilities within your enterprise. It also integrates quickly and easily into your existing infrastructure. You can embed Alphablox Alerts into other applications, including your Alphablox Analytics solutions and third party portals.

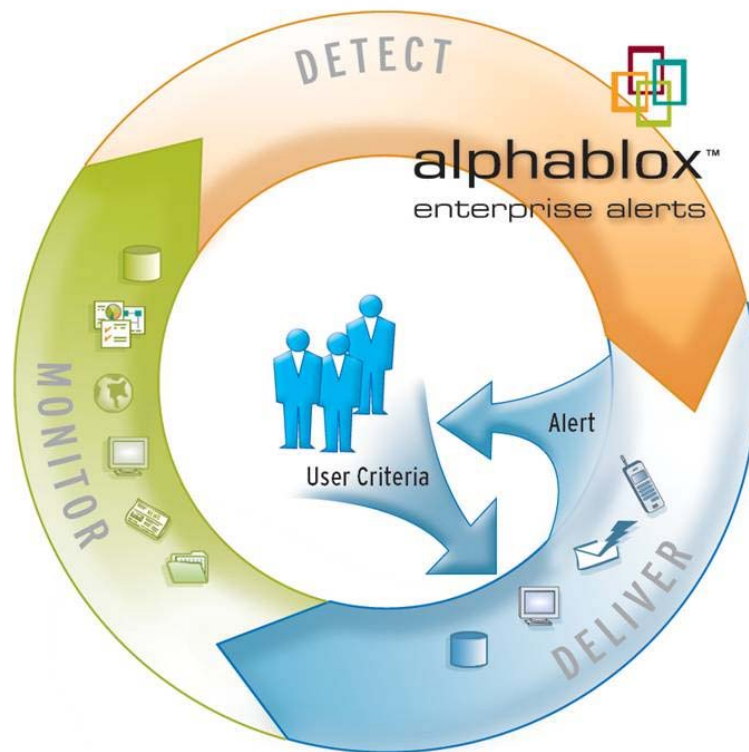
This document will cover the following topics:

- The need for proactive monitoring capabilities in today's business environment
- High-level architecture of Alphablox Alerts
- The components of an alerting application
- The process of building an Alphablox Alerts application.
- The tools for developing an application with Alphablox Alerts.

Readers with knowledge of standard Web technologies, application design and development, J2EE application servers, and JavaServer Pages technology will gain maximum value from this paper.

Introduction

Competitive pressures require businesses to have a clear vision of their market and operations and the ability to adapt rapidly to change. However, true vision and clarity only happen when key business information is delivered to the right people at the right time. An essential component of this process is a powerful, flexible enterprise analytic solution that can accurately deliver information in real time, providing that vision, facilitating change, and creating a competitive advantage.



Alerting--immediate notification of crucial changes in key business metrics -- is now an indispensable part of an analytic solution. Proactive notification based on internal, structured business metrics is critical to timely decision making. Alerts on data from external sources and from unstructured data such as the data from news feeds or web sites can provide additional key insights. Only with Alphablox Alerts can you create alerts based on internal, external and unstructured data. These sources can be combined to create alerts that enable you to quickly analyze the impact outside events may have on your company.

What if you could:

- Alert sales executives when opportunities above certain dollar values change status and when there are relevant news stories about their prospects?
- Alert marketing executives when competitor information changes or becomes available?
- Alert retail buyers when a competitor lowers its price on an item and there is a resulting impact on sales, and provide the buyer the option to change the price they are charging?
- Alert account support managers about critical, new support issues? Or issues that have not received proper attention?
- Alert customers and prospects with new information about specified products and giving them the option to order immediately?
- Accelerate the planning and budgeting process by notifying managers of changes to their budgets and reminding them when their approval is required?

Alphablox Alerts enables these scenarios and others through its flexible monitoring and delivery capabilities. Decision makers decide what information they want and when they want it. They can choose the frequency with which they are notified, as well as any contextual information they require as part of the notification. Application developers have the tools to quickly create efficient monitors that gather data of interest to users and personalize the notifications that are delivered.

Alphablox Alerts

Alphablox Alerts continuously monitors identified, crucial sources of data, looking for potentially significant changes. It matches those changes to criteria specified by subscribed users who are interested in the information and delivers timely notifications through multiple channels selected by the subscribers. These channels include e-mail, wireless messages, pagers, and portals.

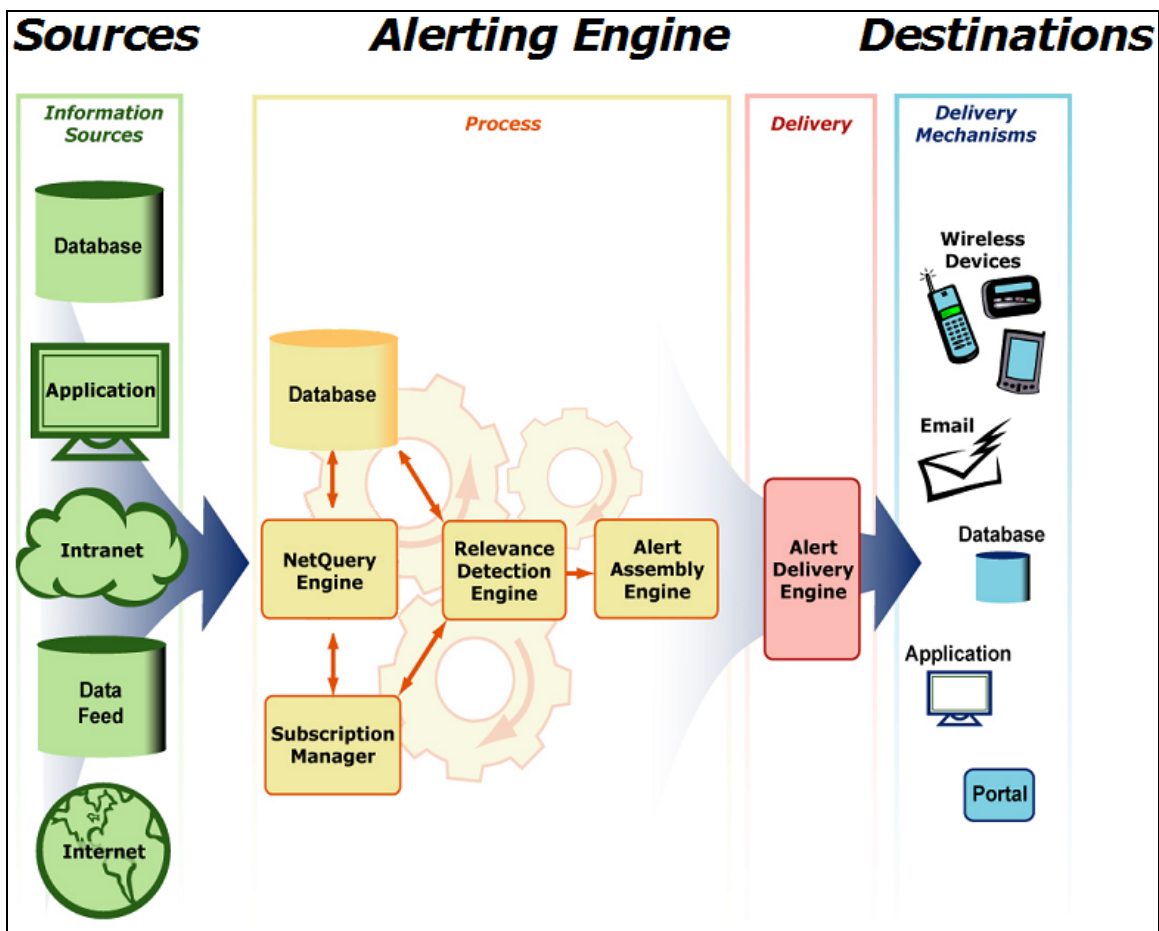
Whereas the majority of alerting products only monitor structured data and look for exceptions – one of Alphablox Alerts' key strengths is its ability to set up complex alerts based on not only structured but also unstructured information.

By leveraging a rich development library, you can build highly customized solutions and integrate them seamlessly into your business environment. Applications built using Alphablox Alerts enable end users to describe the exact circumstances under which they should receive an alert and how the alert should be delivered.

Alphablox Alerts consists of several major processes that perform the following functions:

- Continually monitor events
- Analyze these events for relevance
- Personalize any notifications
- Aggregate and assemble alerts
- Deliver these alerts to the appropriate users and devices

The following diagram shows a high level architectural view of Alphablox Alerts:



- The **NetQuery Engine** continually monitors up to the minute data for changes, based on a flexible set of criteria.
- Besides allowing users to specify when and how alerts are to be delivered, the **Subscription Manager** provides a custom interface with which users specify the events, criteria, and timeliness in which they wish to be alerted.
- The **Relevance Detection Engine** continually compares a user's subscription parameters with the latest monitored events.
- The **Alert Assembly Engine** is responsible for organizing and aggregating alerts into the appropriate format.
- Finally, the **Alert Delivery Engine** forwards the alert response to the appropriate device.

Alphablox Alerts can monitor a variety of data sources, including:

- Relational databases
- Multidimensional databases
- Unstructured textual data (e.g. HTML or XML)

In addition, data may be pushed into the platform using an API.

Typically, the end user enters subscription criteria through user interfaces provided by application developers. Alphablox Alerts stores the user's personalized set of criteria using a proprietary expression based language call Subscription Relevance Language (SRL)[™] that allows the specification of highly complex criteria. In addition to the criteria, the end user may specify how, where, and when she wishes to receive an alert once her criteria have been satisfied.

Once the end-user's criteria are matched, an alert is generated according to the user's preferred format. Many options exist including:

- HTML or plain text Email
- Wireless devices, such as mobile phones and PDAs
- Relational database – An alert may be set up to deliver data into a relational database.
- Custom – An easy-to-use interface is available to deliver Alerts to 3rd party applications.
- Portal or JSP page delivery – Alerts can be stored in a repository and retrieved at a later time via an API for display within a JSP or portal.

Although not a requirement, most Alphablox Alerts applications are web-based. Alphablox Alerts supports industry leading J2EE application servers such as IBM's WebSphere and BEA's WebLogic.

The Alerting Process

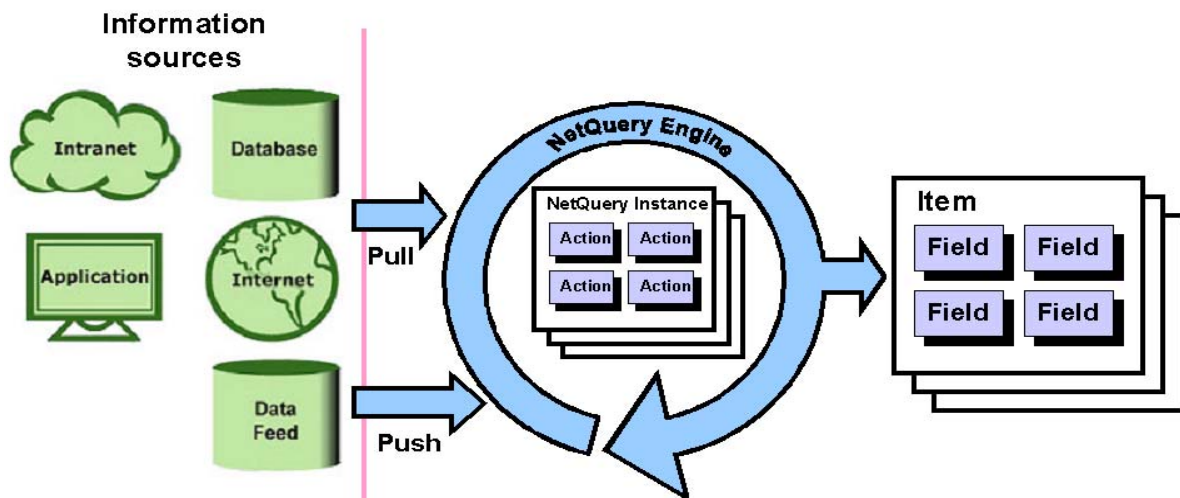
The creation of an alert is a three-step process:

- Collecting interesting data
- Matching the data against user criteria
- Delivery of notifications for data that meets the criteria

Collecting the Data – NetQuery Technology

Alphablox Alerts uses the NetQuery technology to periodically gather data that may lead to potential matches. The application developer instructs the NetQuery engine exactly what data to collect and when. These instructions are packaged into units called NetQueries. A simple NetQuery reads the rows of a table from a relational database. Another might read the output of a query that is run against a multidimensional database. More complex NetQueries target more than one data source.

The NetQuery is instructed to retrieve interested data from the data source being monitored and package the returned data into units called Items. An Item is composed of a collection of name-value pairs called Fields[†]. This is analogous to a row in a relational database table. More than one of these Items may be generated per NetQuery run.



The NetQuery Engine

[†] In addition, a Field contains some basic type information.

A NetQuery is usually scheduled to run periodically, but it can also be initiated by an external event. In addition, data may be pushed into a NetQuery rather than pulled, such as from an external application.

Data sources the NetQuery engine may access fall into two broad categories:

1. Structured data sources
 - a. Relational Database
The NetQuery Engine can access any JDBC 2.0 compliant relational database. Each row of the result set generates an Item.
 - b. Multidimensional Database
Alphablox Alerts currently supports Microsoft Analysis Services and Hyperion Essbase XTD Deployment Services. The NetQuery executes a report against a particular cube or data source. Each row of the report generates an Item.
2. Unstructured data sources
 - a. Textual Data
Alphablox Alerts provides an advanced text parsing language called Extensible Regular Navigation and Indexing Expression (ERNIE)[™]. An application developer can use ERNIE to parse textual data and retrieve information of interest. For example, a developer could use ERNIE to parse a page of HTML to extract an article's information, such as title, date, summary, author, and more.
 - b. POP3 Email
A NetQuery may be configured to retrieve email messages from a POP3 email account. The contents of the email can then be used for potential matching.

A NetQuery may be configured so it remembers the value retrieved from the last run. Since end users are primarily interested in receiving alerts when data is changed or when new data is found, a NetQuery can filter out data already processed.

Matching the Data – Subscription Relevance Language

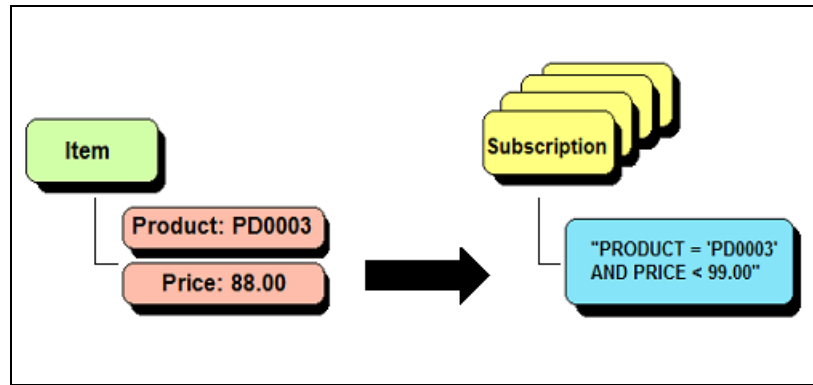
Items returned by a NetQuery execution are fed into the Relevance Detection Engine to be compared against user criteria. As mentioned earlier, criteria are stored on the server as an SRL[™] expression.

Here is an example of an SRL expression. It is very similar to an SQL WHERE clause:

```
PRODUCT = 'PD0003' AND PRICE < 99.00
```

Recall that an Item is composed of a collection of name-value pairs called Fields. A Field name is referenced directly in the SRL expression in much the same way as a column name is referenced in SQL. The SRL expression above expects the incoming Item to contain the Fields "PRODUCT" and "PRICE".

Once the values from a NetQuery Item are substituted into the expression, a true or false value is generated. A true value signifies a match, and a false value signifies the user's criterion was not satisfied.



The values of the Item are substituted into the SRL expression

In addition to the usual mathematical and logical operators, SRL contains advanced textual searching capabilities such as identifying if separate words appear in the specified sequence.

Typically, the subscription page is a JSP that collects user's alerting criteria. The JSP then collects that information, forms an SRL expression, and calls the appropriate API method adding the criteria to the Relevance Detection Engine. To facilitate the process, a JSP tag library is available which streamlines adding criteria to the Relevance Detection Engine from a JSP.

Format and Delivery of the Alert – Alert Templates

Once the Relevance Detection Engine makes a match, an alert is generated and forwarded to the Alert Assembly Engine for delivery. Alphablox Alerts gives the application developer complete control over the format and the schedule of delivery for the alert. Additionally, developers can choose to give the end user control over these variables.

The platform supports various delivery options:

- SMTP – used for email & Short Message Service (SMS) delivery.
- Database – stored in a relational database.
- Portal – stored in the Alerts repository for later retrieval. This enables an alert to be viewed from web applications as portlets.
- Custom – gives the developer control over where the alerts are delivered. This option can be used for integrating Alerts with other applications or services.

Delivery formats are specified using a template. A template is a JSP which, when executed, builds the notification according to the specifications of the developer. Alphablox Alerts does provide some templates as part of its installation, such as a template for generating emails.

In addition to specifying the delivery type, the developer (or end user) can specify the schedule for delivery. Alerts can be delivered as they are found, or consolidated into regularly scheduled deliveries. Consolidation of alerts is important as helps reduce the total number of alerts an end user receives. For example, the end user may request a consolidated alert once a day. In addition, the user may specify blackout periods for a particular destination, e.g. “do not send alerts to my cell phone after 6pm.”

The consolidated matches are passed to the appropriate template. It is the template which generates the final alert. For email delivery, the alert would output HTML. For database delivery, the template would output SQL statements. Usually the template generates the appropriate output and passes it on for final delivery.



Example Alert – HTML email

The screenshot above is an example of an alert end users might receive. The HTML output is generated entirely by an alert template. The template has full access to the details of the matched Items, including the criteria information.

Designing an Alerting Application

Alphablox Alerts was designed to allow a high degree of customization. Everything that the end user sees may be customized to suit his requirements. Application developers who create an Alphablox Alerts application will need to answer the following questions:

- What are the data monitoring requirements?
Data is periodically collected by the platform, and evaluated for potential matches. The developer must decide which data the end users are potentially interested in. A common initial mistake is collecting too much data and limiting system resources.
- What information must be collected from the end user?
Usually, the end user is presented with an easy to use subscription form. It collects the information needed to establish the user's criteria and delivery preferences. It is the application developer that uses this information to create an SRL expression.
- What are the delivery requirements?
How does the end user wish to be notified? When does he wish to be notified? What information should be contained in the notification?

Developing an Alerting Application

Based on the answers to the above questions, the application developer is ready to begin developing an alerts application. In this section, we will delve into more detail regarding the development of an alerting application using Alphablox Alerts.

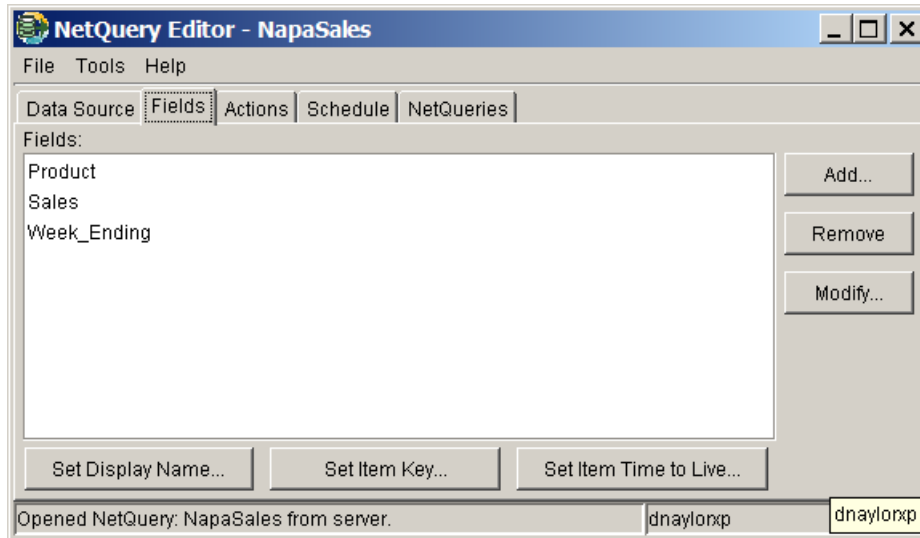
The first step in writing an alerting application is deciding what data you wish to alert on. The platform must be instructed to periodically collect data from various data sources. A developer will accomplish this by writing a NetQuery.

The second step is to develop a web application that collects end user preferences and adds this information to the Alphablox Alerts Server. This section will explain how a web application is created, and cover the resources available that make this task simpler.

The third step is to write the alert templates. This section will show you an example of an alert template, and the JSP tag library that makes template development easy.

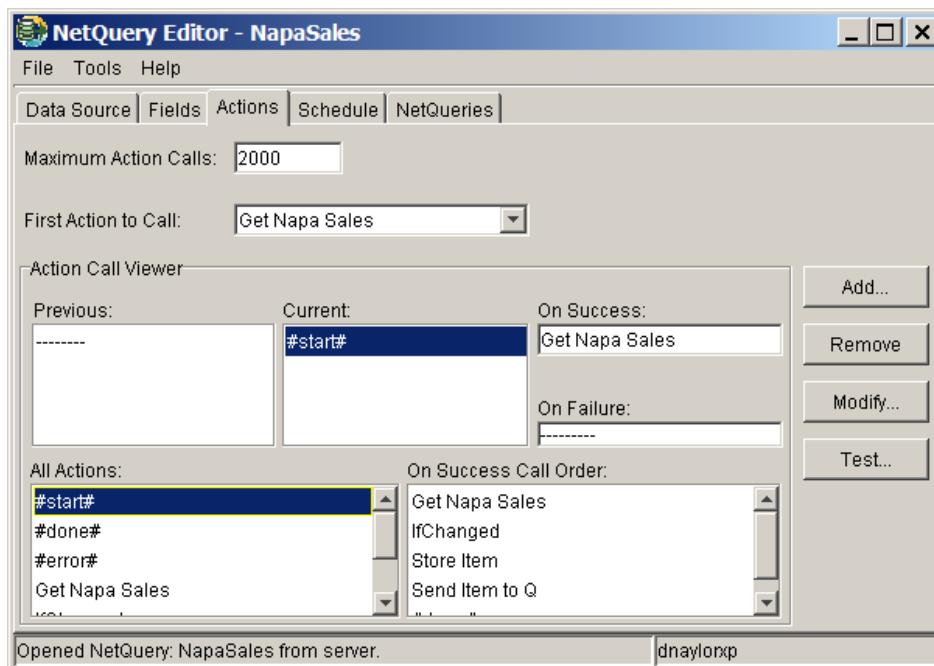
Writing the NetQuery

The NetQuery Engine uses NetQueries to retrieve data. A NetQuery is a set of instructions that the NetQuery Engine follows to collect data from one or more data sources. NetQueries are created using the NetQuery Editor tool.



NetQuery Editor - Fields Tab

A run of a NetQuery generates a collection of Items, each being a collection of name-value pairs called Fields. Once the structure of the Item has been determined, the developer configures a series of instructions (called actions) that tell the NetQuery Engine how to gather the data.

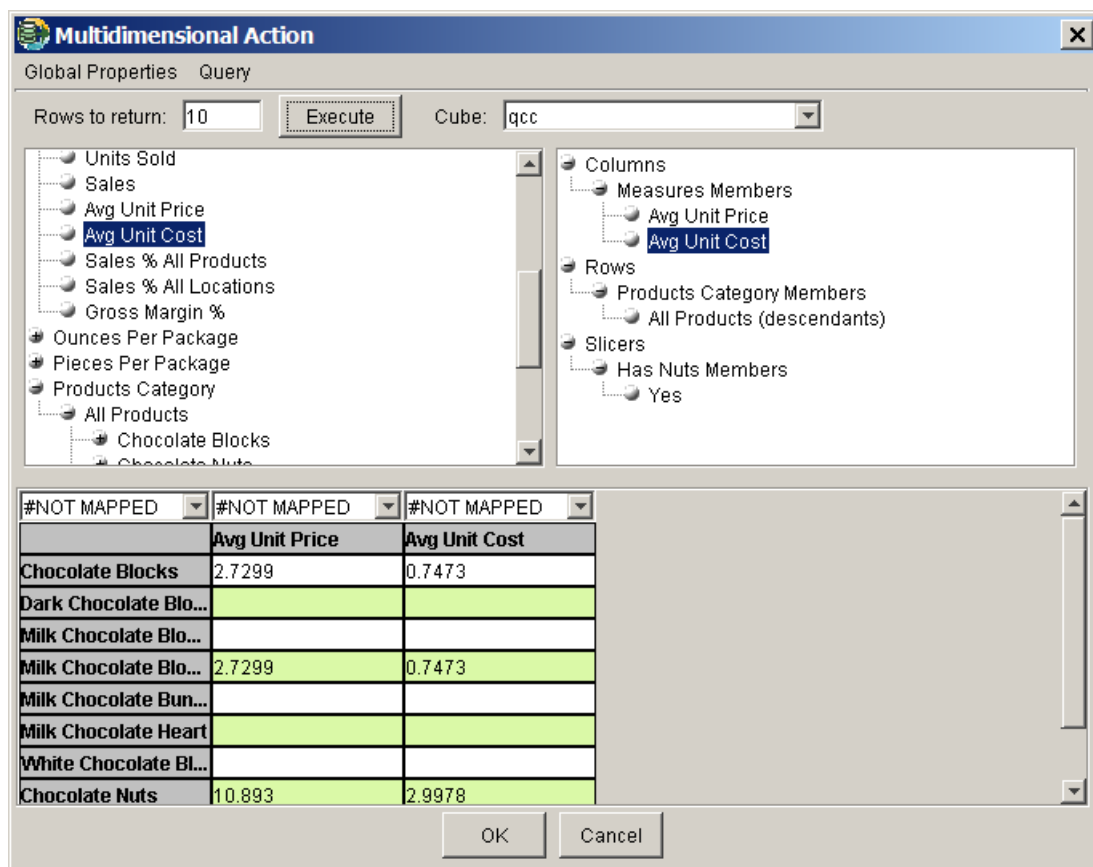


NetQuery Editor - Actions Tab

Many different types of actions are provided by Alphablox Alerts. Examples include:

- Retrieve data from a relational database.
- Retrieve data from a multidimensional database.
- Fetch a page of textual data from a web server.
- Extract a field from a page of textual data.
- Determine if an Item has changed from the last run.

Each action has its own dialog with different configuration parameters. The following screenshot shows the configuration dialog for retrieving data from a multidimensional database:



This action retrieves data from a multidimensional database.

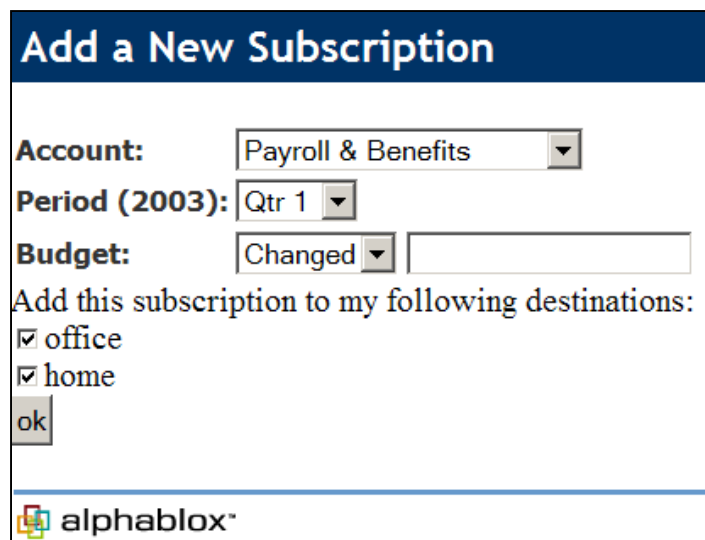
Once the NetQuery has been written and tested, it is then deployed on the Server. The developer may give the NetQuery a schedule that defines how often the NetQuery is run.

Writing the Web Application

The application developer writes the user interface that allows the end users to specify their criteria and destination preferences. Alphablox Alerts wraps the criteria and the destinations it should be linked to into an object called a subscription. Although many technologies may be used, the most common practice is to write a web application that interfaces with Alphablox Alerts.

Alphablox Alerts provides an extensive JSP tag library that greatly simplifies the process of writing a form that collects preferences from the user. In addition, Alphablox Alerts provides a Java API for implementing more advanced tasks, such as adding new users and associating them with subscriptions.

The following screenshot shows a simple HTML form that was written using the JSP tags:



Example Form to Collect Criteria

A section of the code that generated the form is shown below:













```
<ba:subscriptionForm id="form" action="controller.jsp" method="post" >
  <table>
    <tr>
      <td class="stepLabel">Account:</td>
      <td>
        <ba:selectList id="account" >
          <ba:option default="true" selectedValue="Payroll & Benefits" >Payroll & amp;
Benefits</ba:option>
          <ba:option selectedValue="Employee Development" >Employee
Development</ba:option>
          <ba:option selectedValue="Recruiting" >Recruiting</ba:option>
          <ba:option selectedValue="Admin Fees" >Admin Fees</ba:option>
          <ba:option selectedValue="Professional Fees" >Professional Fees</ba:option>
          <ba:option selectedValue="T & E" >T & amp; E</ba:option>
        </ba:selectList>
      </td>
    </tr>
  </table>
  ...
```

JSP Segment

Tags are provided to implement the most common HTML form controls such as radio buttons, checkboxes, selection lists, and edit fields. These controls store user choices and this information is passed on as criteria when the subscription is added to the server.

In addition to the tags that collect preferences from the user, there are subscription definition tags that take information from the subscription UI and create the subscription in the Alphablox Alerts server. These tags take user provided criteria and convert it into SRL expressions.

In addition to writing the subscription form, the developer also adds all the supporting parts of the application such as login, user creation, destination creation, and listing subscriptions. Comprehensive examples are provided that detail how to do this. In many cases, the developer may choose to simply reuse the code provided in the examples. Here is an example UI showing a list of subscriptions:

Current Subscriptions	Edit Schedule Active Delete
Payroll & Benefits,Qtr 1:Budget >1111	   
My News Subscription	   
Alphablox Enterprise Alerts,johnd,5 days	   

Listing Subscriptions

A web application may also allow a user to choose to view his alerts within the application or portal. A user may opt to have his alerts available within the application or portal in addition to his regular email alerts.

My Alerts

Current Alerts

New!	Apr 28, 2003	2003 Income Statement alert for the Electrical division - Marketing.
New!	Apr 28, 2003	2003 Budget Change:BusinessUnit=Microprocessors,Month=Apr,Account=Equipment M...
New!	Apr 28, 2003	2003 Budget Change:BusinessUnit=Microprocessors,Month=Apr,Account=Equipment R...
Seen	Apr 28, 2003	2003 Budget Change:BusinessUnit=Microprocessors,Month=Apr,Account=Equipment M...

Report **2003 Budget Change** contains the following information:

BusinessUnit	Microprocessors
Month	Apr
Account	Equipment Maintenance
Budget	94098 (From 95042)
Department	110 - Facilities

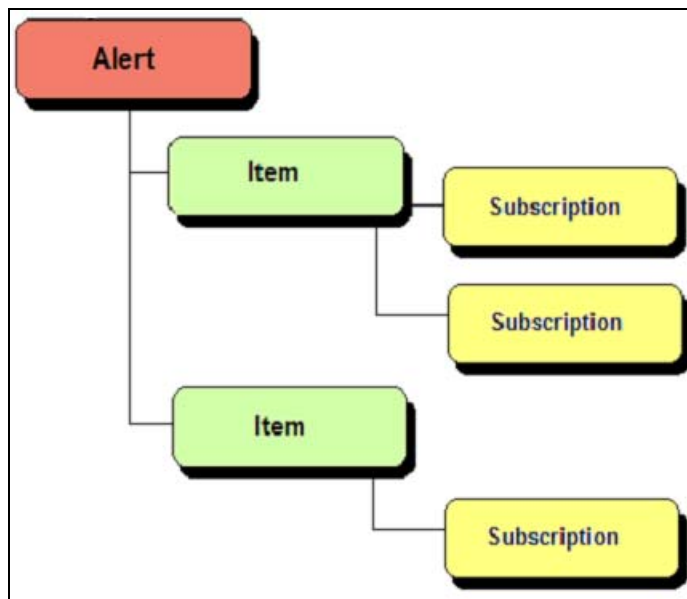
These report values matched the following subscription(s):
2003 Budget Change, Apr

Alerts being shown within the web application

Designing the Alert template

As mentioned earlier, alerts can be delivered in a number of different formats. For each format the developer intends to support, a template must be provided. An alert template is a JSP page created with the help of Alphablox Alerts' custom JSP tags. When the Relevance Detection Engine finds a match, the information is passed to the appropriate alert template. The alert template will then generate output according to its associated format. For example, an HTML alert template will generate an HTML page, while a Database template will generate SQL statements. Sample alert templates are provided that can be used out of the box or customized for your applications.

The alert template will receive its data in a particular format called an alert object.



The Alert Object

Each Item in the alert corresponds to a match made by the Relevance Detection Engine. The alert template mechanism provides a JSP tag library that allows the developer to traverse this object, generating output in the process.

```
<pd:loopItems>
  <pd:loopFields field="field">
    ...
  </pd:loopFields>
  <pd:loopSubscriptions>
    ...
  </pd:loopSubscriptions>
</pd:loopItems>
```

Basic structure of an Alert template

The code segment above shows how the developer would use the tags to iterate through the information contained in the alert.

The following example shows how an alert template intersperses these tags with HTML output:

```
<!-- *****PAGE TITLE***** -->
<table border=0 cellpadding=5 cellspacing=1 width="100%">
  <tr class=pageTitleBg>
    <td class=pageTitle>New Alert!</td>
  </tr>
</table>

<pd:loopItems>
<!-- *****START ARTICLE***** -->
<table border=0>
  <pd:loopFields>
    <tr>
      <td class=messageHead ><pd:fieldName /></td>
      <td class=messageBody><pd:field/></td>
    </tr>
  </pd:loopFields>
  <pd:loopSubscriptions>
    <!-- *****MATCHED CRITERIA***** -->
    <tr>
      <td class=matched>Matched criteria: <font class=matched>
<pd:subscriptionName/></font></td>
    </tr>
  </pd:loopSubscriptions>
</table>
<hr>
</pd:loopItems>
```

Segment from an HTML alert template

The above example might produce an alert similar to the following:

New Alert!	
HoldType	ALL
Count	6
Operator	9999998
Matched criteria: Count > 2	

Example HTML alert

Summary

Alphablox Alerts can be used either standalone or in conjunction with a larger analytic solution to deliver critical information. Only with Alphablox Alerts can you monitor and correlate both structured and unstructured data and deliver personalized notifications to decision makers, providing them critical visibility into the business.

Using the Alphablox Alerts' robust development environment, application developers can build highly customized applications for alerting that integrate seamlessly with your business environment. Application developers can specify:

- What data to collect, and how to collect it
- Complex criteria using SRL
- The front-end web application
- The format of the alert via alert templates

Application developers can give the end users complete control over the various elements of the alert. Users can use customized user interfaces to quickly specify their critical criteria. They can also specify when they would like to be notified, how they would like to be notified and what information should be contained in the notification.

Alphablox Alerts powerful monitoring, detection and notification capabilities, combined with a development environment that enables application developers to quickly deliver customized solutions, means that Alphablox Alerts can provide immediate benefit within your business analytic infrastructure.

About Alphablox

Alphablox Corporation provides the leading InLine Analytics platform enabling Global 2000 corporations to develop and deploy real-time analytic solutions for optimizing business performance. Combining best-in-the-industry professional services with award-winning analytics technology, Alphablox's solutions integrate seamlessly into existing business processes and enterprise infrastructures, empowering front-line business people with personalized analytic applications to drive targeted business objectives. Market leaders in financial services, telecommunications, manufacturing, and retail such as ABN Amro, Deutsche Bank, Blue Cross Blue Shield, Harley Davidson, Mentor Graphics, Neiman Marcus, Seagate, and Toys R Us, rely on Alphablox's powerful analytic solutions. The company has financial backing from leading investment firms, including Accel Partners, Mohr, Davidow Ventures, Integral Capital Partners, and Deutsche Bank, among others. Founded in 1996, Alphablox is headquartered in Mountain View, California, and has offices and affiliates throughout the United States, United Kingdom, Australia, and distributors worldwide. For additional information on Alphablox and its solutions call 888/256-9669 or visit the company's Web site at www.alphablox.com.



Copyright © 1996 - 2003 Alphablox Corporation. All rights reserved.
Alphablox, Alerts, Pearl Diver Platform, NetQuery, ERNIE, Blox, SpreadsheetBlox, and the Alphablox logo are trademarks or registered trademarks of Alphablox Corporation.
All other product names are trademarks of their respective owners.

Tel: 1.888.BLOX.NOW
(1.888.256.9669)
Email: info@alphablox.com
Web: www.alphablox.com

Worldwide Headquarters
516 Clyde Avenue
Mountain View, CA 94043
Tel: 650.526.1700
Fax: 650.526.1701

Alphablox Corporation Limited
Centurion House
London Road, Staines, TW18 1AX

United Kingdom
Tel : +44.1784.410.409
Fax: +44.1784.410.333

Alphablox Australasia
266 Auburn Road
Hawthorn, VIC 3122

Australia
Tel: +61.3.9810.6300
Fax: +61.3.9810.6363