iDashboards

PRODUCT MANUAL

iDashboards Alerts Admin Manual

Version 9.6

iDashboards Alerts Admin Manual

Version 9.6

No part of the computer software or this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from iDashboards. The information in this document is subject to change without notice. If you find any problems with this documentation, please report them in writing to support@iDashboards.com. iDashboards does not warrant that this document is error free.

Copyright © 2004 - 2018 iDashboards. All rights reserved.

Trademarks:

The iDashboards logo and tagline are trademarks of iDashboards.

All other products and company names referenced herein are the trademarks of their respective owners.

Support information:

iDashboards
900 Tower Drive, 4th Floor
Troy, MI 48098

Phone: (248) 528-7160
Fax: (248) 828-2770

Email: support@iDashboards.com
Web site: http://www.iDashboards.com

Table of Contents

1. OSKAR 4

2. INTRODUCTION 4

2.1 Architectural Overview 5
2.2 Overview 6

3. ALERT SERVER INSTALLATION 8

3.1 Windows Installation 8
3.2 Manual Installation 8
3.2.1 Deploying idbalerts.war 8
3.2.2 Choosing a Context Root 8

4. ADMINISTERING ALERTS VIA APPLICATION SERVER CONSOLE 9

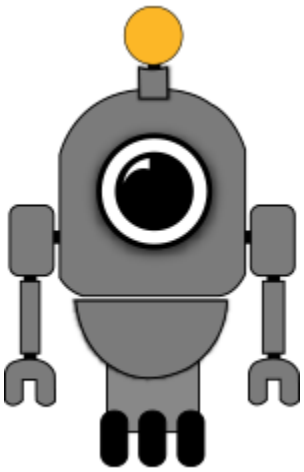
4.1 Determining the URL of the iDashboards Administrator Application 9
4.2 Administration Console 10
4.2.1 Controlling Permissions 10
4.2.2 Browser Alert Checks Enabled 10
4.2.3 Importing and Exporting Charts with Alerts 10

5. ADMINISTERING ALERTS VIA ALERTS SERVER CONSOLE 12

5.1 Server Status 12
5.1.1 Pausing and Restarting the Server 13
5.1.2 Understanding Server Events 13
5.1.3 Troubleshooting Error Events 14
5.1.4 Event Retention 15
5.1.5 Email Events 15
5.1.6 Log Configuration 16
5.2 Alert Administration 16
5.2.1 Retrieving an Alert 17
5.2.2 Modifying an Alert 18
5.2.3 Importing and Exporting Charts with Alerts 18
5.3 System Configuration 18
5.3.1 Modifying a System Setting 18
5.3.2 System Settings 19
5.3.3 System Logs 21
5.3.4 Managing Severity Levels 24
5.3.5 SMS Carrier Costs 27
5.3.6 Configure the SMS Carriers 27

INDEX 30

1. OSKAR



OSKAR, the Online Support & Knowledge Acquisition Repository, is the preferred support resource for iDashboards' customers, partners and prospects. The OSKAR Support Portal can be used to submit, review and update support tickets.

https://oskar.idashboards.com/

Those who have an active, support and maintenance contract with iDashboards also have access to the following content in our User Community:

- **Knowledge Base** – Numerous product and technology articles for your review.
- **Community** – Forums and discussion groups for customers to discuss various topics and products amongst themselves.
- **Resources** – Many downloadable resources that can be used with iDashboards.
- **Ideas** – Area for customers to submit feature requests and great product ideas.
- **Blog** – Thoughts, stories and ideas on data and dashboards

2. Introduction

2.1 Architectural Overview

iDashboards Alerts is a separate server from the iDashboards Server. It is an optional server if using the manual installation process. Like the iDashboards Server, the Alerts Server is J2EE web application, packaged in a WAR file.

Figure 2-1 provides an illustration of the Alerts Server's deployment environment. The Alerts Server connects to the iDashboards repository database, which it reads and updates during its operation. It also connects to the external data sources configured by the iDashboards Server, for the purpose of reading and examining chart data.

An external SMTP service, such as Microsoft Exchange Server or UNIX Sendmail, is used for sending notification emails. This component is optional, however; the Alerts Server can operate without sending emails.

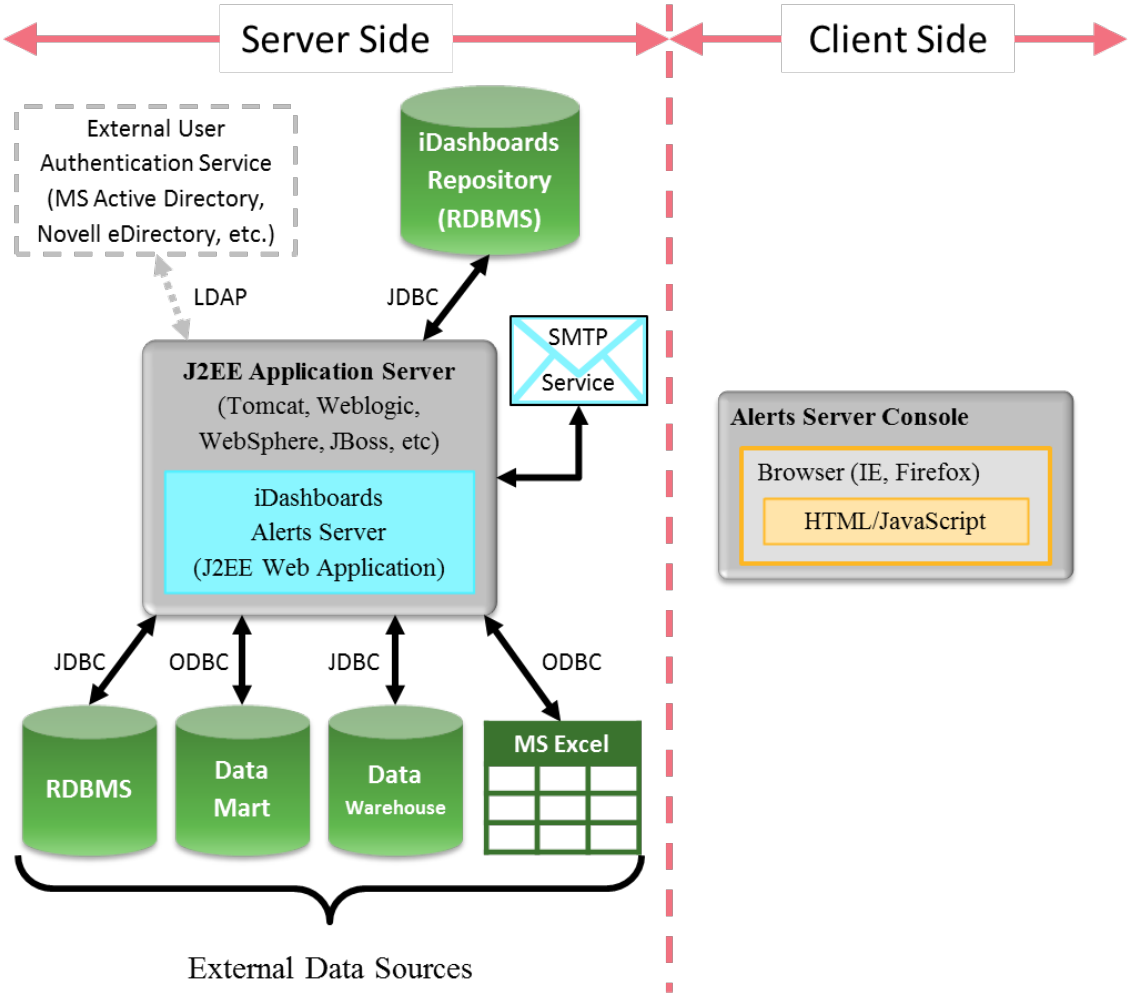


Figure 2-1

2.2 Overview

iDashboards offers real-time alerts which can automatically trigger notifications via email. Alerts can be triggered based on a variety of thresholds, trend-based conditions and other criteria. When an alert is triggered, users can receive additional information at the time of the alert to assist with faster root cause analysis and problem resolution.

iDashboards Alerts is a server process which monitors chart data and awaits for the moment when a certain condition is encountered (ex. exceeding a threshold, results are blank, etc.). Once the condition is met, an alert is sent to specific iDashboards users via on-screen notification and/or email. Alerts are configured with a monitoring schedule, and will monitor the chart data even if nobody is viewing a dashboard.

Note: On-screen alerts are available via the Flash User Application not the HTML User Application

The term “alert” can have different meanings in different contexts. At a general level, an alert is a mechanism that notifies iDashboards users that certain conditions exist within chart data. The term “alert” is sometimes used to refer to the notification itself, i.e. the item appearing in a user’s alerts dashboard. It is also used to describe the configuration stored in the repository that defines the conditions for an alert, its name and the message that is displayed to users when the conditions are met.

Alert Terminology:

- **Alert** – Unless its context suggests otherwise, the term “alert,” as used in this manual, will refer to the configuration of an alert – the conditions, name, severity level, message text, etc. – that is stored in the iDashboards repository database.
- **Check** – An alert is “checked” by the alerts server according to a predefined schedule. This means that the alerts server loads the data of the chart for which the alert was configured, and evaluates it according to the alert’s rules.
- **Trigger** – If, during an alert check, the alert's rules are satisfied by the chart data, then the alert is said to be “triggered”.
- **Instance** – When an alert triggers, an “instance” of the alert is created. It is this instance that appears in the alerts dashboard of the iDashboards User Application.

The primary components of alerting in iDashboards involve:

- **Designing an alert** – Alerts can be configured on any chart and must be associated to a chart. You cannot have an alert without a chart. This task involves identifying the condition(s) needed to trigger an alert.
- **Scheduling an alert**– Determine the frequency the alert should check for a condition. Administrators should assist with determining an appropriate schedule since alert checking utilizes server performance.
- **Determining the audience** – Alerts can be for personal use or for groups of iDashboards users.

- **Resetting an alert** – Think ‘snooze button’. Once an alert has triggered, how much time needs to pass before checking the condition again.

3. Alert Server Installation

iDashboards Alerts requires an additional installation that should occur on the same server where iDashboards is installed. For the installation to take place, administrative access to the server is necessary. Access to the installation media for iDashboards may also be necessary to retrieve certain files.

3.1 Windows Installation

The Alerts Server is automatically installed when using the iDashboards .EXE executable during the Application Server installation.

3.2 Manual Installation

Note: *The installation process assumes that iDashboards has already been fully installed.*

Note: All installation files are located in the bin directory on the iDashboards installation media.

The exact procedure for installing iDashboards Alerts may vary from one installation to the next. The process normally involves copying a WAR file into a specific directory and then restarting the server.

Note: *The location of the ivizgroup home directory is displayed on both the login screen and the home screen of the iDashboards Admin application. If the ivizgroup home directory is on a UNIX or Linux server, you should log in as the user account under which the iDashboards server will be running when making any changes inside the ivizgroup home directory. This will insure that any files or subdirectories that are created will be readable and writable by the iDashboards server.*

3.2.1 Deploying idbalerts.war

- Locate the file “idbalerts.war” and copy it into the application server deployment folder. Another option includes uploading the WAR file through the application server using a web interface.

```
Example: …\Tomcat\webapps\idbalerts.war
```

3.2.2 Choosing a Context Root

Regardless of the application server used to host the Alerts Server, it must be assigned a “context root” within the server’s URL space. Conceptually, the context root can be thought of as a subdirectory beneath the server’s root URL, which forms the root of the web application’s URL space. This allows multiple web applications from different sources to be deployed to the same application server without URL conflicts.

It is recommended that “/idbalerts” be used as the context root for the iDashboards web application. Since the Alerts Server WAR file is named idbalerts.war, some application servers (such as Tomcat) will automatically default its context root to “/idbalerts”, so choosing that can simplify the deployment process.

4. Administering Alerts via Application Server Console

The Alerts server console has a unique URL; however this section discusses the Alert settings within the Application Server Console. All of the Alerts administration described in this chapter is performed in the iDashboards Administrator's Application.

4.1 Determining the URL of the iDashboards Administrator Application

The URL of the Administrator Application is the web address used to access settings with a web browser. To determine it, you must know three things:

1. The hostname or IP address of the server on which iDashboards is deployed.
2. The port number on which the iDashboards application server is listening, if it is other than port 80, which is the default HTTP port number.
3. Refer to the iDashboards Admin manual to understand the function of the "context root". Normally this will be "idashboards."

Once these three components are known, the URL of the Administrator Application will be:

```
http://<servername or IP address>:<port number>/<context root>
```

For example, if the hostname is "dashmachine", the port number is 8080, and the context root is "idashboards", the URLs for iDashboards would be:

Interface	URL
Flash User Application	`http://dashmachine:8080/idashboards`
Desktop Application	`http://dashmachine:8080/idashboards`
HTML User Application	`http://dashmachine:8080/idashboards/html5`
Application Server Console	`http://dashmachine:8080/idashboards/admin`
Reports Server Console	`http://dashmachine:8080/idbreports`
► Alerts Server Console	`http://dashmachine:8080/idbalerts`
If the port number is 80, it can be omitted from any of the URLs, as seen below:	
Application Server Console	`http://dashmachine/idashboards/admin`

In some cases, such as when the server is being accessed over the Internet, it may be necessary to append a domain name to the hostname:

```
http://dashmachine.mycompany.com/idbalerts
```

When the Server Console URL's are accessed through a web browser, a login screen should appear.

4.2 Administration Console

One Alert setting can be found within the iDashboards Administrator's Application when iDashboards Alerts is installed. Note that this is different from the Alerts Administrator Application which is accessed through a different URL.

4.2.1 Controlling Permissions

A user must have access to a chart before configuring an alert on a chart. Access to dashboards is controlled through the permission settings for groups and categories.

4.2.2 Browser Alert Checks Enabled

Navigate to System > System Settings and then select "User Application Settings" from the category option (see Figure 4-1).

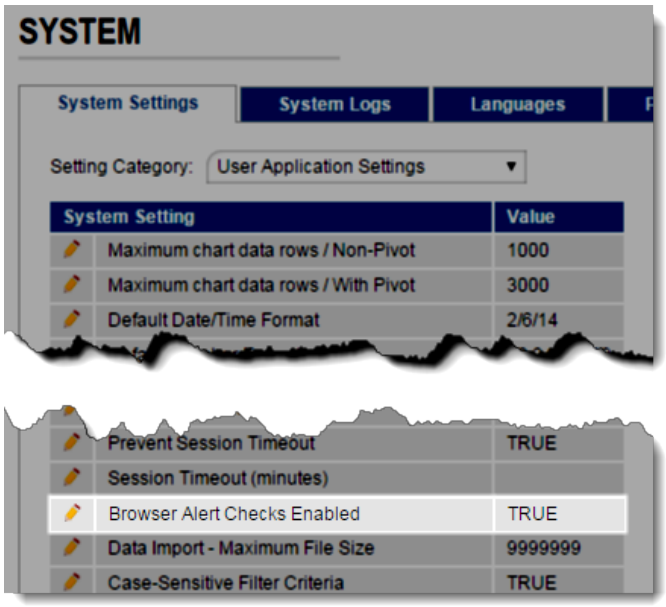


Figure 4-1

This setting determines whether or not browsers will contact the iDashboards server periodically to check for new alerts. The default setting is TRUE; however it can be set to FALSE when the Alerts server is offline for an extended period of time, to reduce the load on the iDashboards server. This setting is sent to a user's browser upon login, so changing it will have no effect on active login sessions.

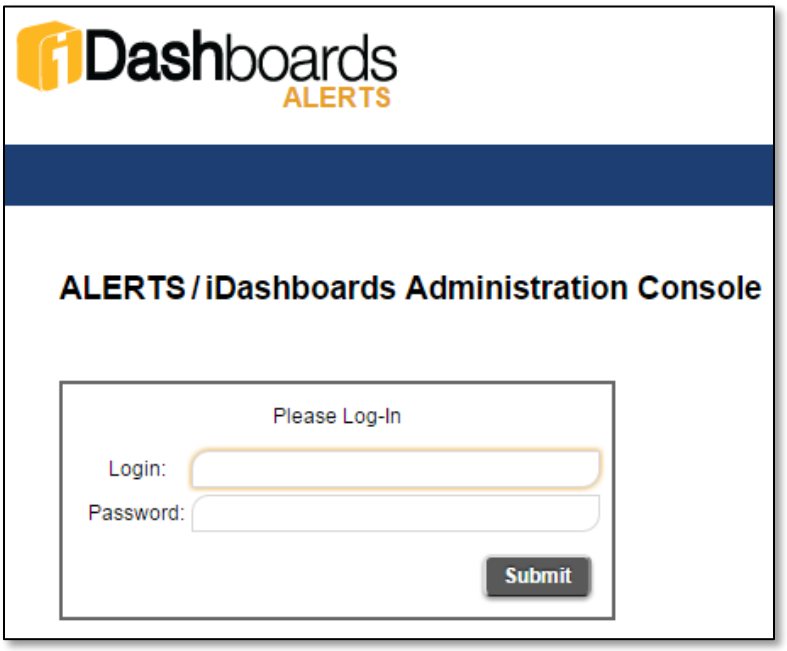
4.2.3 Importing and Exporting Charts with Alerts

Charts and dashboards can be exported from one iDashboards repository and imported into another iDashboards repository. This process is only performed through the iDashboards Administrator Application and is detailed in the iDashboards Administrator's Manual.

Charts that contain alerts will retain their alerts when exported if the alerts are not private. Private alerts will not be exported. Also, specific group notifications will not be preserved as iDashboards will not reconcile groups. The user will have to edit the alert after it has been imported and decide which groups will be notified

5. Administering Alerts via Alerts Server Console

Multiple settings of the Alerts Server can be controlled through the system configuration screens of the Alerts Server console.



5.1 Server Status

Within the regular iDashboards Server, nothing much happens unless a user or administrator does something in a browser that causes a request to be sent to the server, like opening a dashboard or saving a chart. Otherwise, it sits idle, waiting for user input.

The Alerts Server is different. Even when there are no administrators logged in, the server can be busy, checking alerts, reacting to a triggered alert, sending emails, etc. If an error occurs on the regular iDashboards Server, it is usually in response to some user action, and is immediately noticed by the user. Within the Alerts Server, however, an error can occur at any time and go unnoticed, and as a result, alerts might fail to generate when they should.

The Alerts Server provides a “window” through which its inner workings can be observed. This window is the Server Status screen of the Alerts Server console. To access the screen, click “Server Status” in the application menu. This screen will show a list of server events (see Figure 5-1).

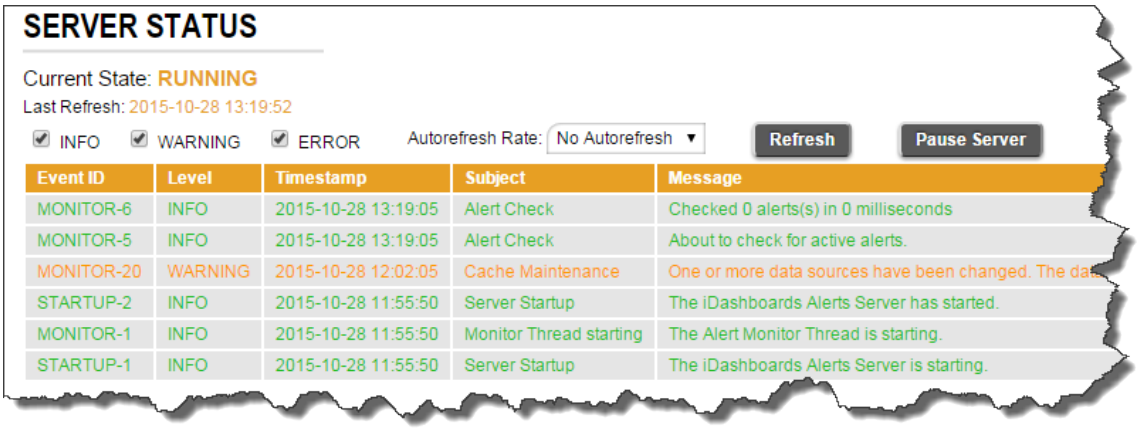


Figure 5-1

5.1.1 Pausing and Restarting the Server

At any given moment, the Alerts Server will be in one of two possible states:

- **Running** – In this state, the Alerts Server is performing all of its normal activities, such as alert checks, sending emails, etc.
- **Paused** – In this state, the Alerts Server does not perform activities such as alert checks or sending emails, however, the Alerts Server console is still fully functional.

In its default configuration, the Alerts Server enters the running state when it is started. When it is in the running state, the Server Status screen will display the line, “Current State: RUNNING”, and the rightmost button (referred to herein as the toggle button) will be labeled “Pause Server”. A running server can be paused by clicking the toggle button.

When the Alerts Server is in the paused state, the Server Status screen will display “Current Status: PAUSED” and the label on the toggle button will say “Start Server”. It can be placed back into the running state by clicking the toggle button.

Normally, the Alerts Server should be left in the running state. The paused state is generally only useful when performing troubleshooting or certain configuration changes.

5.1.2 Understanding Server Events

The most prominent feature of the Server Status screen is the list of server events. A server event can be any type of noteworthy occurrence, such as the server being paused or a database error. The event list can be filtered to only display events of certain, selected levels. This is accomplished by checking or unchecking the checkboxes for the different event levels.

A server event has the following attributes:

5.1.2.1 Event ID

Each server event is assigned a code referred to as the “event ID”, which identifies the type of event that it is. And event ID consists of an event category, such as “MONITOR”, and a number, separated by a hyphen.

The event category is used to identify approximately where in the system the event occurred. For example, the MONITOR category is for events that occur on the monitor thread, which is the main thread that runs continually inside the server, checking alerts and performing other tasks.

The number portion of the event ID uniquely identifies the type of event within an event category. For example, “MONITOR-7” is the event ID used to indicate that a routine alert check occurred.

5.1.2.2 Level

Each server event has one of the following three levels:

- **INFO** – This level is used for routine events. INFO-level events are displayed in green text in the event list.
- **WARNING** – This level is for events that occur during normal operation, but should be noted by a server administrator. WARNING-level events are displayed in the yellow text in the event list.
- **ERROR** – This level is used for abnormal, unexpected events such as a database error that occurs during alert generation. ERROR-level events are displayed in red text in the event list.

5.1.2.3 Timestamp

The event timestamp is the date and time at which the event occurred.

5.1.2.4 Subject

The event subject is a short phrase describing the event.

5.1.2.5 Message

The event message is a short sentence that contains information about the event.

5.1.3 Troubleshooting Error Events

In some cases, an ERROR-level event displayed in the event list may contain hyperlinks to other screens that display additional information about the error or the associated alert.

For example, if the event’s level code, “ERROR”, is followed by “(!)” the exclamation mark can be clicked on to display the Java stacktrace that was generated by the error.

Although stacktraces appear undecipherable to almost anyone other than Java developers, they usually provide clues as to the cause of an error. For example, a stacktrace notice might read “Connection timed out”, indicating that the Alerts Server was unable to connect to the SMTP service to send emails.

A stacktrace can also be copied and pasted into an email to support@idashboards.com to assist iDashboards support staff in troubleshooting errors.

When an error is associated with a specific alert, the event message in the list may be followed by “(View Alert)”. This is a hyperlink that can be clicked to open the administrative screen for the errant alert, through which the alert can be temporarily disabled or permanently deleted.

5.1.4 Event Retention

During normal operation, the Alerts Server is frequently recording new events in the event list. Because of this, one would expect that over time, the event list would grow extremely large, yet it does not. This is because only a certain number of events with a given event ID are retained in the event list. This number is referred to as the "retention depth" for that event ID. When the number of events with a particular event ID exceeds the retention depth for that ID, the oldest ones are removed from the list and discarded, keeping the entire event list at a manageable size.

The retention depth for an event ID is normally not of concern to the Alerts Server administrator. It can be viewed, however, by holding the mouse cursor over the event ID in the event's list. This will produce a tool tip, similar to the one shown in Figure 5-2, displaying the retention depth for the event ID.

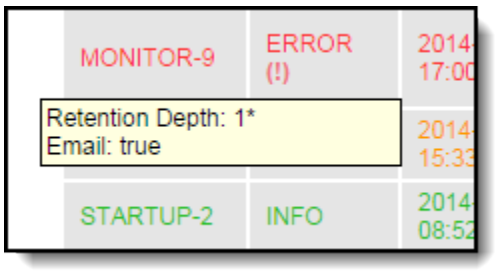


Figure 5-2

5.1.4.1 Qualified Event Retention

For some error events, the retention depth is not applied to the event ID alone, but rather to the event ID combined with some hidden qualifying information. For example, if the error event is related to a particular alert, that alert's ID number might be used as the qualifying information. So, if the event ID is "MONITOR-9" and the alert ID is 123, the hidden, "qualified" event ID to which the retention depth would apply would effectively (if not actually) be "MONITOR-9-123". And if the retention depth for MONITOR-9 events is 1, that really means that one MONITOR-9 event related to alert #123 will be retained in the list, but at the same time a MONITOR-9 related to alert #905 might be retained in the list as well. This keeps important events from being pushed out of the event list before they can be viewed by an administrator.

If a retention depth applies to a qualified event ID as described above, it will be followed by an asterisk (*) in the event ID tool tip.

5.1.5 Email Events

Certain event types are designated as "email events". When an email event occurs, a notification email will be sent to the designated Alerts Server administrators, provided that:

- The Alerts Server is properly configured to send event notification emails.
- The level of the event (INFO, WARNING, or ERROR) is at or above the configured threshold at which the event notification emails are sent.

To determine whether or not an event in the list is an email event, hold the mouse cursor over its event ID until the tool tip appears. It will include the line “Email: true” for email events, and “Email: false” for non-email events.

5.1.6 Log Configuration

At runtime, the Alerts Server will log system errors and other events in a log file. The name of the log file is idbalerts.log, and it will be created in the <IVIZGROUP HOME>\logs directory. Certain parameters can be set in the ivizgroup.properties file to determine the maximum size a log file will be allowed to grow to, the number of backups that will be kept, and the verbosity of the logging output. Note that these settings can be changed while the server is running through the System Logs screen, however such changes will not persist across a server restart.

log.directory – This property can be used to indicate a directory other than <IVIZGROUP HOME>\logs where log files should be written. It must exist and be writable by the iDashboards application server process. Forward slashes (/) should be used instead of backslashes (\) as a path separator.

log.maxFileSize – This property indicates the maximum size, in bytes, that a log file will grow to before it is "rolled over", that is, renamed with a ".1" extension so that a new idbalerts.log file can be created. This property must be an integer from 0 to 9,223,372,036,854,775,808. (Do not include commas.) The suffixes "KB", "MB", or "GB" can be appended to indicate the value is kilobytes, megabytes or gigabytes, respectively. If no value is given, the default used is "10MB".

log.maxBackupIndex – When logs are rolled over, the current idbalerts.log file is renamed to idbalerts.log.1, an existing log file with a ".1" extension is renamed with a ".2" extension, one with a ".2" extension is renamed with a ".3" extension, and so on up to the value of the log.maxBackupIndex property. If a log file already has an extension equal to log.maxBackupIndex, it is discarded when the log files are rolled over. If log.maxBackupIndex is zero, there will be no backup files, and the log will be truncated when its size grows to the maximum size.

log.level – This value must be one of the following: ERROR, WARN, INFO or DEBUG. (The default is INFO). DEBUG will produce the most verbose output, and ERROR will produce the least. Normally, DEBUG should only be used when troubleshooting.

5.2 Alert Administration

Alerts are normally created and maintained through the iDashboards User Application as described in the User Manual. The Alerts Server console, however, provides options through which limited modifications can be made to existing alerts, specifically:

- Alerts can be enabled or disabled. When disabled, an alert is not checked by the Alerts Server.
- Email notifications can be enabled or disabled for individual alerts.

- Alerts can be deleted.

Administrative access to alerts is independent of the iDashboards security framework. An administrator can perform the above modifications on any alert in the system, regardless of the category to which the alert belongs, or whether the alert is public or private.

The alert administration screens are accessed by clicking ALERTS in the application menu. This will display the Alerts search screen, shown in Figure 5-3.

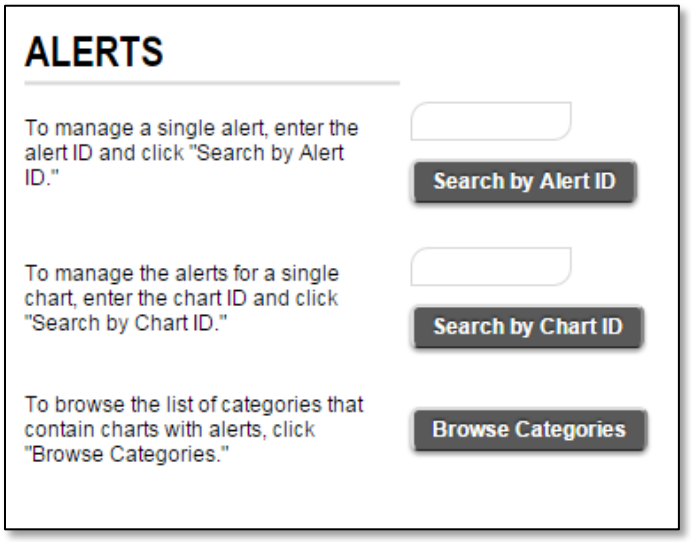


Figure 5-3

5.2.1 Retrieving an Alert

Before an alert can be modified, it must first be retrieved from the repository. If the Alert ID number is known, it can be retrieved directly. If the ID number of the alert's associated chart is known, the alert can be selected from a list of alerts associated with that chart. If neither the alert ID nor chart ID is known, then the alert can be "browsed" for.

5.2.1.1 Searching by Alert ID

The Alert ID is the number that uniquely identifies an alert in the iDashboards repository. To retrieve an alert with its ID, enter the ID in the appropriate text box. If the alert with the given ID exists in the repository, it will be displayed on the screen.

In the iDashboards User Application, the Alert ID is visible on the summary panel of the configuration dialog (see Figure 5-4)

Figure 5-4

5.2.1.2 Searching by Chart ID

The Chart ID is the number that uniquely identifies a chart in the iDashboards repository. Enter the Chart ID in the appropriate text box. If the chart exists and has one or more alerts configured, the chart alerts will be displayed.

Note: In the iDashboards User Application, the Chart ID is visible within the Chart Designer.

5.2.1.3 Browsing for an Alert

To browse for an alert, click the button labeled “Browse Categories”. This will display a list of categories in the iDashboards system that has charts with alerts. The total number of alerts within each category will also be displayed. Clicking the “View Alerts” link for a category will display a list of chart alerts within the category, along with other details.

5.2.2 Modifying an Alert

Administrative modifications can be made to an alert through the Alert Administration screen. Once an alert is located, certain properties of the alert can be edited. However, some details of the alert can only be edited through the User Application.

5.2.3 Importing and Exporting Charts with Alerts

Charts and dashboards can be exported from one iDashboards repository and imported into another iDashboards repository. This process is only performed through the iDashboards Administrator Application.

Charts and Dashboards that contain alerts will retain their alerts when exported if the alerts are not private. Private alerts will not be exported. Also, specific group notifications will not be preserved as iDashboards will not reconcile groups. The user will have to edit the alert after it has been imported and decide which groups will be notified.

5.3 System Configuration

Various aspects of the Alerts Server can be controlled through the system configuration screens of the Alerts Server console. The system configuration screens are accessed by clicking SYSTEM on the application menu

5.3.1 Modifying a System Setting

The four categories of system settings are:

1. Alerts
2. Notification Email Settings
3. SMTP Settings
4. SMS Settings

To modify a system setting, click its Edit icon ( ). For most system settings, the value must be entered into a textbox, while for others, the value can be selected from a dropdown list. The edit form for a system setting will include a description of that setting and its valid values.

After a system setting has been modified, click the “Save” button to save the changes, or “Cancel” to discard the changes and return to the system settings screen.

5.3.2 System Settings

System settings are global settings used to control the Alert’s Server’s behavior. They are managed through the System Settings screen, which is the first screen displayed when SYSTEM is clicked on the application menu. The System Settings screen can also be accessed by clicking the “System Settings” tab that appears on the other system configuration screens.

System settings are grouped according to their function into *setting categories*. A dropdown list of the available setting categories appears near the top of the System Settings screen. When a category is selected, a list of the system settings for that category appears on the System Settings screen.

5.3.2.1 Configure the Alert Settings

Notification email settings and SMTP settings are discussed in other chapters of this manual. The settings in the Alerts category are discussed below.

5.3.2.1.1 Server Startup State

This setting determines the initial state of the server upon startup. The two possible values are:

1. **Running** (default) – The Alert Monitor Thread will be started, and the server will check for alerts according to its schedule.
2. **Paused** – The Alert Monitor Thread will be in the paused state when the server starts up. It will need to be started manually through the STATUS screen of the Alerts Admin application.

5.3.2.1.2 Alert Instance Retention (Days)

This setting indicates the number of days an alert instance will be kept before it "ages out" of the alert queue and is deleted from the repository database. Allowable values are from 1 to 9999. If the setting is left blank, then alert instances will remain in the queue indefinitely.

Note: This setting will not remove or alter alert configurations. The default is null.

5.3.2.1.3 Browser Alert Check Interval (Minutes)

This setting indicates the interval, in minutes, in which a user’s Alerts dashboard will check for new alert instances. Allowable values are from 1 to 60. The default is 1 minute.

5.3.2.1.4 Maximum Displayed Alert Instances

This setting indicates the maximum number of alert instances that will be displayed in a user's Alerts dashboard. If the number of alert instances for a user exceeds this maximum, the newest ones will be given priority. Allowable values are from 20 to 200.The default is 50 instances.

5.3.2.2 Configure the Notification Email Settings

The iDashboards Alerts Server is capable of sending emails in response to certain events. The three types of emails sent are:

- **Alert Notifications** – An alert can be configured so that an email is sent to its recipients when the alert is triggered.
- **Server Event Notifications** – These emails are sent to a predefined list of email addresses (which presumably belong to server administrators) when certain routine (non-error) server events occur, such as the startup or shutdown of the server.
- **Server Error Notifications** – These emails are sent when certain types of errors occur on the server, such as a database error during an alert check. They are sent to the same email addresses that receive server event notifications.

5.3.2.2.1 Email Configuration Roadmap

For the alerts server to send email notifications, it must first be properly configured. The overall steps to accomplish this are:

- **Configure the SMTP (Simple Mail Transfer Protocol) Settings** – Notification emails are sent through an external SMTP service, such as UNIX Sendmail or Microsoft Exchange Server. The Alerts Server must be configured with enough information to connect to, and if necessary, authenticate itself to the SMTP service.
- **Configure the Notification Email Settings** – These settings include information such as the name and email address used in the “from” header of outgoing emails, the list of email addresses that will receive server event notifications, and the information that is included in the subject lines of notification emails.
- **Configure the Email Templates** – This is an optional step that provides a great deal of control over the information included in the bodies of notification emails. Using email templates, notification emails can be sent in both HTML format (including images) and plain text. If this step is omitted, emails will be sent as plain text and include only a minimal amount of default information.

5.3.2.3 Configure the SMTP Settings

As mentioned previously, the iDashboards Alerts Server uses an external SMTP service to send emails. On the SMTP Settings screen, locate the following settings:

- **SMTP Host** – This is the hostname or IP address of the machine on which the SMTP service is running.
- **SMTP Port** – This is the number of the TCP/IP port on which the SMTP service is listening. (The standard SMTP port number is 25.)

- **SMTP Service Requires Authentication** – Set this to ‘Yes’ if the SMTP service requires authentication or incoming connections, or to ‘No’ if it does not.
- **SMTP Service User** – If the SMTP service requires authentication, this setting must contain the username of the user that will be used to connect to it, otherwise it should be left blank.
- **SMTP Service Password** – If the SMTP service requires authentication, this setting must contain the password that will be used to connect to it, otherwise it should be left blank.
- **SMTP Encryption** – This setting determines the type of encryption (if any) used to secure the connection with the remote mail server. The options are ‘None’, ‘SSL (Secure Socket Layer)’ and ‘TLS (Transport Layer Security)’.

5.3.2.4 Configure the SMS Settings

An SMS text message can also be used to notify users when an alert has triggered.

- **SMS Notifications Enabled** – If this setting is "No", then all SMS notifications will be disabled. If it is "Yes", then SMS notifications will be sent for alerts that have SMS notification enabled, to users that have an SMS phone number configured and elected to receive SMS notifications.
- **Maximum Number of Segments** – This is the maximum number of segments of the complete SMS message that will be sent to the user's phone. Allowable values are from 1 to 99. The default value is 10. Larger values may result in lot of SMS notifications being sent out to accommodate the entire message.
- **SMS Segment Prefix Enabled** – If this setting is "Yes", then each segment of a long SMS message will be prefixed with the index number and total number of segments (e.g. (1/10) indicates first segment out of 10 segments of a long SMS message). If it is "No", then SMS segments will not be prefixed with indices.

5.3.3 System Logs

Log settings are managed through the Log Settings screen. The Log Settings screen can be accessed by clicking the “System Logs” tab on any of the system configuration screens where it appears (see Figure 5-5).

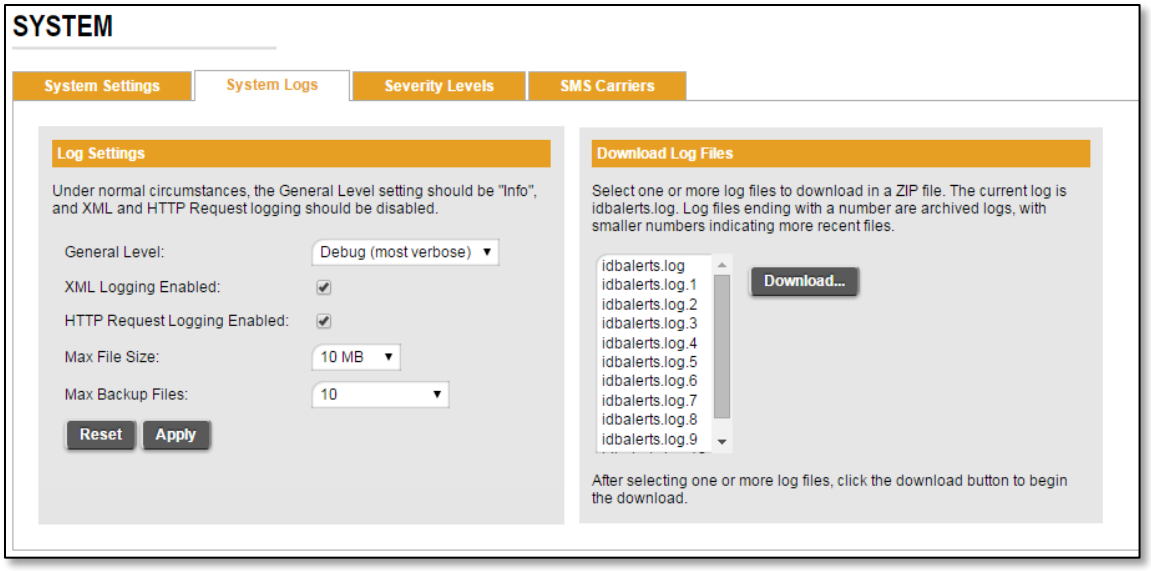


Figure 5-5

When the iDashboards server is started, the initial log settings are read from the ivizgroup.properties file, and if they are not present, the defaults shown in Figure 5-5 are used. Once the server has been started, the settings can be changed through the Log Settings screen, however, changes made will not persist beyond a server restart. Under normal operating circumstances, the settings shown in Figure 3 6 should be used.

Changes made to log settings are not applied until the “Apply” button has been clicked. The available settings are:

5.3.3.1 General Level

This setting determines the types of log messages that will be written to the log file. Each level can be thought of as a threshold, with Debug being the lowest and Error the highest. When a level is selected, all messages categorized at that level and above will be written to the log. The available levels are as follows:

- **Debug (default)** – This is the most verbose setting, and could impact system performance on a busy server. Debug log messages are generally only useful to an iDashboards support representative, so this level should only be used when troubleshooting problems.
- **Info** –This is a far less verbose level than Debug, which writes information about the operating environment to the log when the iDashboards server is started. It is the recommended level for normal operations.
- **Warn** – In addition to error messages, this level will write warning messages about server events that are noteworthy but not critical.
- **Error** – This is the least verbose log level. It will only write messages to the log when a critical error occurs.

5.3.3.2 XML Logging Enabled

When this checkbox is checked, XML that passes between the Alerts Server Console screens and the server is written to the log file. It causes very verbose output which is only useful to an iDashboards support representative, so it should remain unchecked except for troubleshooting purposes. The default is unchecked.

5.3.3.3 HTTP Request Logging Enabled

When this checkbox is checked, information about the HTTP requests that are sent to the Alerts Server from the Alerts Server Console screens is logged. As is the case with XML logging, it causes very verbose output which is only useful to an iDashboards support representative, so it should remain unchecked except for troubleshooting purposes. The default is unchecked.

5.3.3.4 Max File Size

This is the maximum size to which a log file will be allowed to grow before it is overwritten by a new one or archived. The default is 10Mb.

5.3.3.5 Max Backup Files

This setting indicates the maximum number of archived log files that will be kept. When an active log file, named “idbalerts.log” grows to its maximum allowed size, it will be renamed with to idbalerts.log.1 and a new idbalerts.log file will be started. If there is already a file named idbalerts.log.1 in the logs directory, it will be renamed idbalerts.log.2, and so on, up to the maximum number of archived log files. When the maximum number has been reached, the oldest archived log file will be discarded. The default is 10.

5.3.3.6 Downloading Log Files

The active log file (idbalerts.log) and any existing archived log files (idbalerts.log.1, idbalerts.log.2, etc.) can be downloaded through the Log Settings screen. To do so, select the desired files from the list at the right of the screen and click the “Download…” button. The selected files will be bundled into a ZIP file by the server and downloaded.

5.3.3.7 Sending Log Files to iDashboards Technical Support

When working with iDashboards technical support to troubleshoot problems with the Alerts Server, it is useful to provide the Alerts Server’s log file(s) to the support representative. Problems can be diagnosed and corrected more expeditiously if these steps are followed prior to contacting iDashboards technical support:

- Set the General Level to Debug, and enable XML logging and HTTP Request Logging.
- Recreate the error condition through the User Application or the appropriate Administrator Application screen.
- Download the idbalerts.log file, and the idbalerts.log.1 file if it exists.
- Email the ZIP file containing the log file(s), along with a description of the problem (and steps to recreate it if possible) to support@idashboards.com.

5.3.4 Managing Severity Levels

Every alert has a severity level associated with it, which indicates whether the news brought by the alert is good or bad, and to what degree it is good or bad. A severity level is represented by an integer value from 0 to 999. Although the meaning associated with a severity level is determined by the administrator who configures it, the suggested convention is that lower numbers (0-499) should be used to indicate bad news – the lower the number, the worse the news – and higher numbers (500-999) indicating good news – the higher the number, the better the news.

In addition to its integer value, a severity has two other important attributes:

- **The severity name** – a short name, for example “Crisis” or “Monthly Sales Goals Reached”.
- **The severity color** – a color that is displayed on alert notifications.

In its default configuration, the Alerts Server provides four built-in severity levels:

Level	Name	Color
300	Crisis	Red
400	Caution	Yellow
600	Good	Blue
700	Excellent	Green

Table 1

Note: These levels cannot be deleted, nor can the integer value; however their names and colors can be changed.

In addition to the default severity levels, an administrator can add new ones and delete existing (non-built-in) ones.

Severity levels are managed through the Severity Levels screen. To access the Severity Levels screen, select SYSTEM from the application menu, and then click the Severity Levels tab as shown in Figure 5-6.

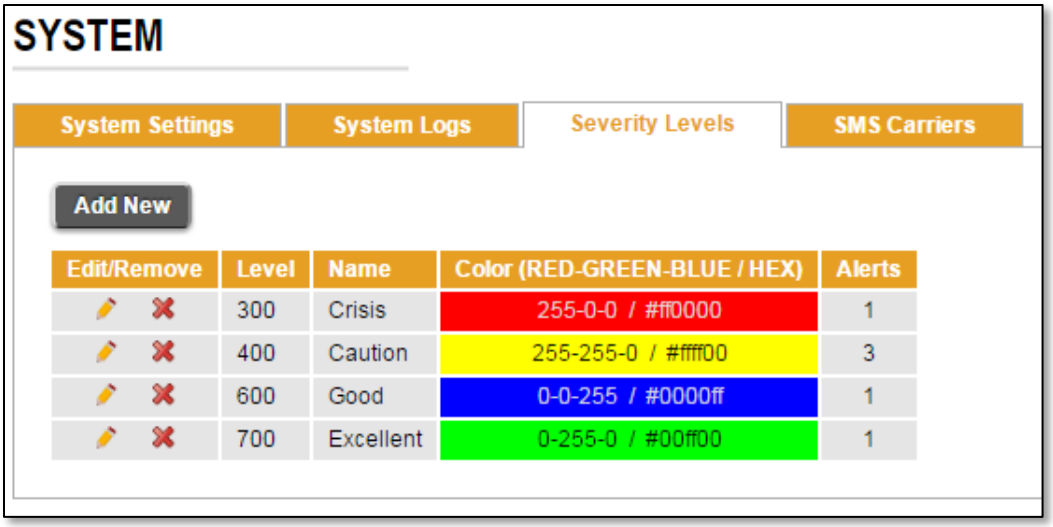


Figure 5-6

5.3.4.1 Adding a Severity Level

To add a severity level click the “Add New” button on the Severity Levels screen. This will open the Severity Level edit screen, shown in Figure 5-6. Enter an integer value from 0 to 999 for the severity level, and a name consisting of from one to 50 characters.

The severity color is defined as a mixture of red, green and blue component colors. Enter an integer value from zero to 255 for each component color, or click and drag on the color’s slider bar to set its value. The severity color will be displayed in the preview box to the right of the slider bars.

After the Severity Level Edit screen has been completed, click the Save button to save the new severity level, or the Cancel button to dismiss the screen without saving it.

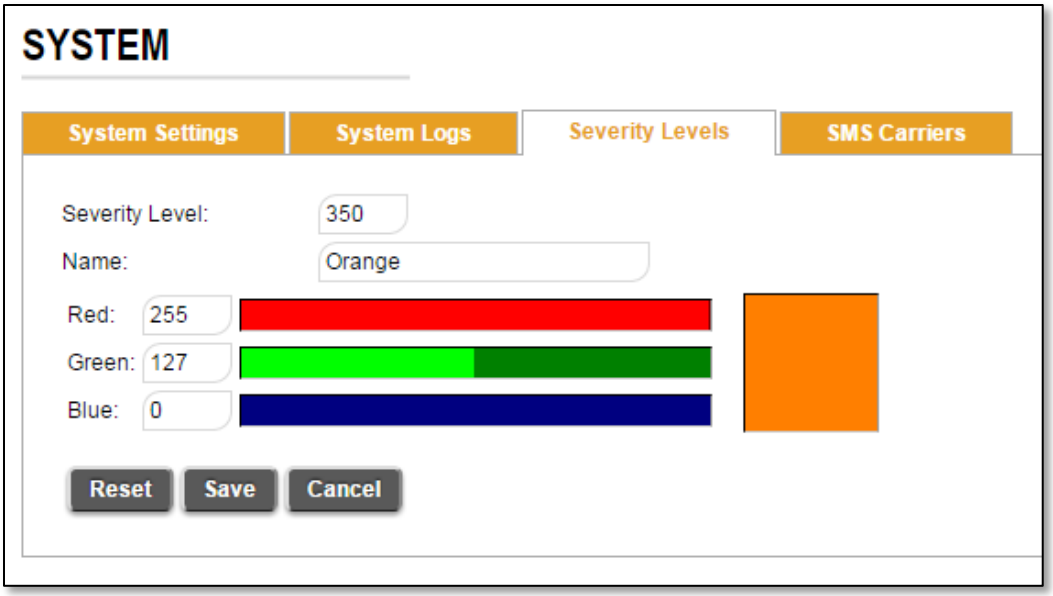


Figure 5-7

5.3.4.2 Modifying a Severity Level

To modify a severity level, click its Edit icon ( ) on the Severity Levels screen. This will open the Severity Level Edit screen, through which the severity level's attributes (other than its integer value) can be modified.

To save the changes, click the Save button, or click the Cancel button to discard the changes and dismiss the screen. Clicking the Reset button will discard the changes, but keep the Severity Level Edit screen displayed.

Note: *Any changes made to a severity level will be visible on any alerts that have a severity level, and all instances of those alerts.*

5.3.4.3 Deleting a Severity Level

Severity levels can be deleted, provided that: **A.** they are not one of the four built-in severity levels shown in Figure 5-7and **B.** no existing alerts are using them as their severity level. The numbers of alerts that are using each severity level are shown in the Alerts column on the Severity Levels screen.

To delete a severity level, click its Delete icon ( ) on the Severity Levels screen. If it is not a built-in severity level, and there are no alerts using it, it will be deleted immediately (without confirmation); otherwise, the operation will fail with a warning message.

5.3.5 SMS Carrier Costs

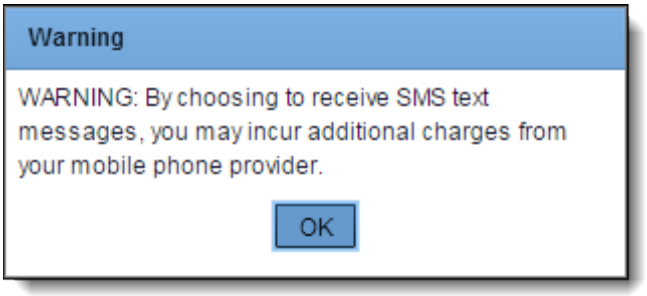
WARNING!
Carrier messaging rates may apply

SMS is essentially the text messaging service offered by all major carriers. The iDashboards Alerts feature has an option to send an SMS text message to users when an alert is triggered. SMS text messages are not enabled by default. The iDashboards administrator and each user partake in setting up each user enrollment in receiving SMS alerts.

Not all users subscribe to SMS or may incur a cost for all SMS messages received. Before using Alerts with SMS, review the SMS capabilities and costs of any iDashboards user configured in the system.

iDashboards assumes no responsibility for fees incurred by utilizing the SMS text messaging feature. Any SMS text messaging fees that are incurred will be billed on 'your' individual mobile provider bill.

When each user enables SMS notifications, through the User Application, they will see the following warning:



5.3.6 Configure the SMS Carriers

Within iDashboards, all major mobile phone provider can be configured with a unique Short Message Service (SMS) set of technical properties. Participating SMS carriers include (but are not limited to): AT&T, SprintPCS, T-Mobile®, Verizon Wireless. Upon installation of Alerts, a variety of popular carriers are available. Existing carriers can be deleted or edited and additional carriers can be added. A carrier already appearing in the list may need to be added more than once to compensate for legacy SMS syntax differences.

Navigate to System > SMS Carriers (see Figure 5-8).

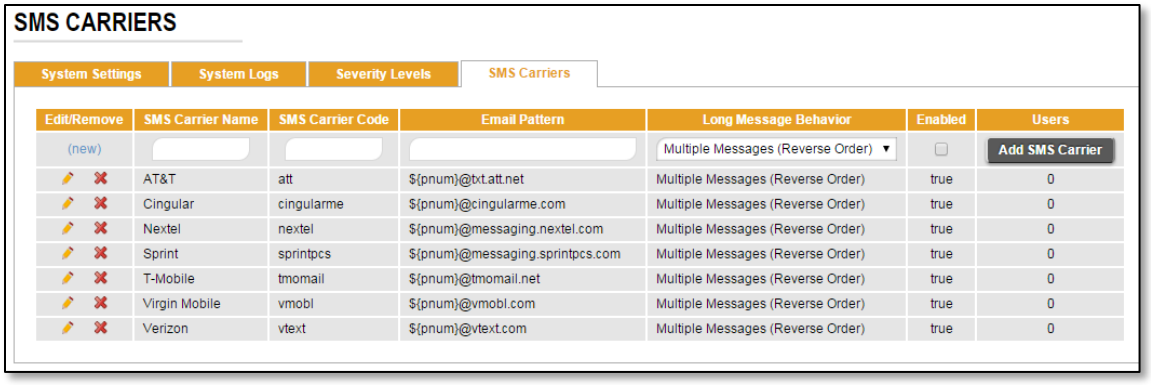


Figure 5-8

5.3.6.1 Edit a Carrier

Any carrier appearing in the list can be edited. To modify a carrier, click its Edit icon ( ). The Carrier code cannot be changed, but all other fields can be updated.

5.3.6.2 Deleting a Carrier

To delete a carrier, click its Delete icon ( ). Carriers can be deleted if the number of associated users is zero, as seen in the right hand column of Figure 5-8. If a carrier has associated users, then the user much dissociate their User Profile with the Carrier. Users can associate or dissociate their account with a single carrier through the User Application.

5.3.6.3 Add a Carrier

All fields are required to add a carrier (see Figure 5-8). Each carrier created will help convert email messages into text messages using a syntax provided by your carrier.

- **SMS Carrier Name** – This field is used to display the SMS carrier name in a friendly format. If a carrier has a legacy carrier code syntax you will need to create multiple entries for the carrier; however, the carrier name cannot be duplicated.
    - **Tip**: If a carrier needs to be entered more than once, try to simply add a numeric suffix (ex. AT&T_2)
- **SMS Carrier Code** – Each SMS carrier has a unique code that is often derived from the Email Pattern below.
- **Email Pattern** – Each SMS carrier has a unique Email-to-Text string that is capable of converting an email message into a text message.
    - **Prefix** - ${pnum}
        - The ‘pnum’ macro stands for phone number. This macro will be programmatically replaced with a users’ 10-digit phone number (ex. 2485551212@txt.att.net)
    - **Suffix -** <email pattern>
        - The first character will be the ‘@’ symbol, followed by the email-to-text string provided by your carrier.

- **Long Message Behavior –** Each carrier has a configuration for handling long text messages. Keep in mind carriers usually have a text message limit of 160 characters.
    - **Send Entire Message –** This option will attempt to send a text message of any character length. If the character length exceeds 160, then the carrier provider will handle message segmentation.
    - **Truncate Message –** This option will ensure the iDashboards text message will be reduced at-or-below 160 characters before sending the message.
    - **Multiple Messages –** This option will have iDashboards segment any messages greater than 160 characters, sending multiple messages if necessary.
    - **Multiple Messages (Reverse Order) <default> –** Same as 'Multiple Messages', but the sending order will be in reverse order (sending the last message segment first, and continuing until the first segment is sent last)
- **Enabled –** This setting will allow for globally enabling or disabling of a carrier in the list. Note: If an entire carrier needs to be enabled or disabled, the administrator may have to enable or disable multiple entries to accommodate for multiple entries associated to legacy syntax.

5.3.6.4 United States Nationwide Popular Carrier Examples

In addition to the popular carriers pre-configured by iDashboards, administrators can add additional carriers using the example shown in Table 2.

SMS Carrier Name	SMS Carrier Code	Email Pattern
AT&T	att	${pnum}@txt.att.net
Sprint	sprintpcs	${pnum}@messaging.sprintpcs.com
T-Mobile	tmomail	${pnum}@tmomail.net
Verizon Wireless	vmobl	${pnum}@vmobl.com

Table 2

Index

A

Alert Checks, 5
Alert Instances, 5
Alerts, 5
  firing, 5

C

Checks. *See* Alert Checks

E

Email
  notification emails, 15

F

Firing. *See* Alerts

I

iDashboards Repository, 5
Instances. *See* Alert Instances

S

Severity Color, 23
  editing, 23
Severity Levels, 5
  adding, 23–24
  deleting, 24–25
  modifying, 24
Support
  OSKAR, 3