

# Rise Alert Application Programming Interface (API)

## Overview

The purpose of this document is to define Rise Alert Application Programming Interface (Rise Alert API).

Goals of the Rise Alert API are as follows:

- Enable 3rd party applications to securely send, cancel and browse active alerts while re-using RDN user account database for authentication purposes.
- Provide compatibility with broad range of platforms and programming languages.
- Provide code samples and the ability for 3rd party developers to test their implementation for API compliance prior to going live.

## Security considerations

In order to support broadest possible range of platforms and programming languages, authentication mechanism is kept intentionally simplistic (username and password must be supplied for every API call; session is *not* maintained) and the security is provided at a transport level (SSL). This allows clients developed using platforms with limited security capabilities (such as Flash) to be used while still maintaining adequate level of protection against attacks.

Only RDN user accounts that are authorized to send alerts on RDN are able to send alerts through Rise Alert API. Currently these are accounts that have roles of Alert Administrator or RDN Administrator assigned to them, but specific RDN roles can be changed in the future releases of RDN.

## API Description

3rd party application developers can choose between two types of Rise Alert API implementation options:

- **XML API** enables 3rd party developers to access all Rise Alert API features. XML implementation is recommended for cases when it is necessary to select display appliances alert is sent to. XML implementation is also recommended for applications that need to provide the ability to view the list of active alerts and cancel specific alerts.
- **HTML API** enables 3rd party developers to send alerts to ALL display appliances in the company using HTTP POST request and name/value pairs in HTML forms rather than XML. HTML implementation is particularly recommended for simple applications that cannot or do not need to generate XML.

## Server URL

Both XML and HTML API are available at the same Rise Alert API server URL:

<https://api.risedisplaynetwork.com/alerts/>.

## XML API

### Guidelines for XML API Requests

The following guidelines explain how to format XML documents that are send in Rise Alert API requests.

- All XML messages between the client and Rise Alert API server must use UTF-8 (Unicode) encoding. UTF-8 encoding should be specified by including the following line at the start of each XML API request:

```
<?xml version="1.0" encoding="utf-8"?>
```

- To include the XML reserved characters &, <, and > in an XML element value, they must be encoded as hexadecimal numeric character references. The following table shows the numeric character references for these characters:

Character	Hexadecimal character reference
&	&#x26;
<	&#x3c;
>	&#x3e;

All other UTF-8 characters can be used directly.

- Username and password must be supplied with all XML requests to the Rise Alert API server using HTTP Basic Authentication. The following instructions explain how to format HTTPS request headers to use HTTP Basic Authentication:

- Create a properly formatted XML file for the command being executed.
- Follow these steps to create the HTTPS header for your request:
  - Create the **Authorization** header for your request.
    - Append a colon to your RDN user name. Then append your RDN password to this value. (The result will have the format **user\_name:password**).
    - Base64-encode the value you produced in the previous step.
    - Provide the **Authorization** header for your request using the following format. Replace the text **Authorization\_Key** in this header with the value you produced in the previous step:  
 Authorization: Basic **Authorization\_Key**
  - Include the **Content-Type** header with the value **application/xml; charset=UTF-8**.

For example, if your user name is **user1** and your password is **password123**, you would base64-encode the value **user1:password123**. The example below shows how the base64-encoded value would appear in your request header:

```
Authorization: Basic dXNlcjE6cGFzc3dvcmQxMjM
Content-Type: application/xml; charset=UTF-8
```

- Send the XML structure created in step 1 using HTTP POST request to Rise Alert API server.

### Getting the list of available display appliances

The list of all available display appliances to which alerts can be sent to can be obtained from RDN by posting the following XML to the Rise Alert API server URL:

```
<?xml version="1.0" encoding="utf-8"?>
<list-appliances />
```

Note: *All available display appliances* are all appliances which belong to the user's company and for which alerts are enabled.

If the request was received and processed successfully, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<appliance-list count="2">
  <appliance>
    <appliance-id>256b4e2e-247c-4220-87f7-0e3a8d7a4d6f</appliance-id>
    <appliance-name>Test Appliance #1</appliance-name>
  </appliance>
  <appliance>
    <appliance-id>95150c2c-5fe8-4a9f-8215-3af5c72e0041</appliance-id>
    <appliance-name>Test Appliance #2</appliance-name>
  </appliance>
</appliance-list>
```

XML Element	Description
appliance-list	Contains a list of <b>appliance</b> elements. Attribute <b>count</b> contains number of <b>appliance</b> elements in appliance-list element.
appliance	Contains information about single appliance.
appliance-id	Contains appliance identifier
appliance-name	Contains appliance name

If the user account is not found, not authorized or not allowed to send alerts the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid username and/or password.</error-message>
</error>
```

XML Element	Description
error	Contains single <b>error-message</b> element.
error-message	Contains error message text.

If the command value is invalid, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid command.</error-message>
</error>
```

If there was a problem on the server, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Server error. Please try again later.</error-message>
</error>
```

### Sending the alert

Alert can be sent by posting the following XML to the Rise Alert API server URL:

```
<?xml version="1.0" encoding="utf-8"?>
<alert>
  <alert-destination all="no">
    <appliance-id>256b4e2e-247c-4220-87f7-0e3a8d7a4d6f</appliance-id>
    <appliance-id>95150c2c-5fe8-4a9f-8215-3af5c72e0041</appliance-id>
  </alert-destination>
  <alert-message>This is a test alert.</alert-message>
  <alert-duration>5</alert-duration>
</alert>
```

XML Element	Description
alert	Contains the definition of alert.
alert-destination	Defines alert destination. Contains the optional list of <b>appliance-id</b> elements. If attribute <b>all</b> equals <b>yes</b> , alert is sent to all display appliances in the company for which alerts are enabled, if attribute <b>all</b> equals <b>no</b> , alert is sent to the display appliances with identifiers matching the list of <b>appliance-id</b> elements.
appliance-id	(Optional) Contains appliance identifier of appliance the alert should be sent to.
alert-message	Contains the alert message (less than 300 characters long).
alert-duration	Contains the duration of the alert in minutes (integer number between 1 and 99).

If the request was received and processed successfully, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<alert-sent>
  <alert-id>321b4e2e-247c-4220-87f7-0e3a8d7a4d6f</alert-id>
</alert-sent>
```

XML Element	Description
alert-sent	Contains the information about alert that has been sent.
alert-id	Contains the identifier of the alert that has been sent. This identifier can be used to cancel the alert.

If the user account is not found, not authorized or not allowed to send alerts the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid username and/or password.</error-message>
</error>
```

If the command value is invalid, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid command.</error-message>
</error>
```

If the **alert-destination** element contains no **appliance-id** elements and its **all** attribute equals **no**, or if the **all** attribute equals **yes**, but the company has no display appliances for which alerts are enabled the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>No destination. </error-message>
</error>
```

If the **alert-destination** element contains invalid **appliance-id** elements and its **all** attribute equals **no**, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid destination. </error-message>
</error>
```

If the **alert-message** element is invalid, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid message.</error-message>
</error>
```

If the **alert-duration** element value was invalid, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid duration.</error-message>
</error>
```

If there was a problem on the server, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Server error. Please try again later.</error-message>
</error>
```

### Cancelling the alert

Alert can be cancelled by posting the following XML to the Rise Alert API server URL:

```
<?xml version="1.0" encoding="utf-8"?>
<cancel-alert>
  <alert-id>ad5e56b1-5561-44df-945c-c3f71a7a5d00</alert-id>
</cancel-alert>
```

XML Element	Description
cancel-alert	Contains single <b>alert-id</b> element.
alert-id	Contains the identifier of the alert to be cancelled.

If the request was received and processed successfully, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<alert-cancelled>
  <alert-id>ad5e56b1-5561-44df-945c-c3f71a7a5d00</alert-id>
</alert-cancelled >
```

XML Element	Description
alert-cancelled	Contains single <b>alert-id</b> element.
alert-id	Contains the identifier of the alert that was cancelled.

If the user account is not found, not authorized or not allowed to send alerts the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid username and/or password.</error-message>
</error>
```

If the command value is invalid, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid command.</error-message>
</error>
```

If the **cancel-alert** element contains no **alert-id** element, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>No alerts. </error-message>
</error>
```

If the **cancel-alert** element contains invalid **alert-id** element, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Invalid alert. </error-message>
</error>
```

If there was a problem on the server, the following XML will be returned:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <error-message>Server error. Please try again later.</error-message>
</error>
```

## HTML API

### Guidelines for HTML API Requests

The following guidelines explain how to format XML documents that are send in Rise Alert API requests.

- HTTP POST request must be used.
- **Content-Type** header with the value **application/x-www-form-urlencoded** must be included in the request.

- Username and password are supplied as part of the form therefore using HTTP Basic Authentication is not required.

### Sending the alert to all display appliances in the company

Alert to all display appliances in the company can be sent by sending the following HTTP POST request to the Rise Alert API server URL:

Form field name	Form field value
<b>username</b>	RDN user name
<b>password</b>	RDN password
<b>message</b>	Alert message to be displayed (less than 300 characters long).
<b>duration</b>	Duration of the alert in minutes (integer number between 1 and 99).
<b>response</b>	<p>(Optional) Determines the server response mode.</p> <p>Possible values are <b>status</b> and <b>page</b> (default: <b>status</b>)</p> <p><b>status</b>: server responds by sending back the appropriate HTTP Status code (200, 400, 401 or 500). No page is returned.</p> <p><b>page</b>: server sends back HTML page which contains user-friendly description of the status and a “Back” button. HTTP Status code is always 200. This mode is primarily intended for debugging/troubleshooting purposes.</p>

Based on the selected response mode, the following responses are returned from the server:

Scenario	response=status	response=page
Alert was sent successfully.	HTTP status code <b>200</b> (OK) is returned.	HTTP status code <b>200</b> (OK) and the following page is returned:  <b>Alert was sent successfully.</b> <input type="button" value="Back"/>
User account is not found, not authorized or not allowed to send alerts.	HTTP status code <b>401</b> (Not Authorized) is returned.	HTTP status code <b>200</b> (OK) and the following page is returned:  <b>Invalid username and/or password.</b> <input type="button" value="Back"/>
Message value is invalid.	HTTP status code <b>400</b> (Bad Request) is returned.	HTTP status code 200 (OK) and the following page is returned:  <b>Invalid message.</b> <input type="button" value="Back"/>
Duration value is invalid.	HTTP status code <b>400</b> (Bad Request) is returned.	HTTP status code <b>200</b> (OK) and the following page is returned:

	returned.	Invalid duration. <input type="button" value="Back"/>
There was a problem on the server that caused the alert not to be sent.	HTTP status code <b>500</b> (Internal Server Error) is returned.	HTTP status code <b>200</b> (OK) and the following page is returned:  Server error. Please try again later. <input type="button" value="Back"/>

## Rise Sandbox Service

Rise Sandbox Service is a development environment that is designed to help 3<sup>rd</sup> party developers test their Rise Alert API implementation. The Sandbox offers the same functionality as the production Rise Alert API system with the following exceptions:

- The Sandbox requires developers to log in using test account.
- The Sandbox does not actually send alerts to displays.
- The Sandbox only supports 2 test appliances: "Test Appliance #1" and "Test Appliance #2"
- The Sandbox only supports 1 alert (same Alert Id = "AAAAAAAA-AAAA-AAAA-AAAA-AAAAAAAA" is returned when alert is sent),

Rise Sandbox Service for Alert API is available at <https://sandbox.risedisplaynetwork.com/alerts> (both XML and HTML APIs).

Test user account for use with Rise Sandbox Service:

User Name	Password
devtest	sandbox

## Code Samples

### XML API

C# sample project is provided (see **sampleCode\CsharpSample** sub-folder). Project and solution files for both Visual Studio 2005 (CsharpSample2005.csproj, CsharpSample2005.sln) and Visual Studio 2008 (CsharpSample2008.csproj, CsharpSample2008.sln) are provided.

### HTML API

Adobe Flash Action Script sample project is provided (see **sampleCode\FlashSample** sub-folder)

Simple HTML page sample is also provided (see **sampleCode\HtmlSample** sub-folder).