



Informiam
Product Line

Informiam Frontline Advisor
Web Service for Alerts

January 15, 2009



Table of Contents

1	Overview.....	2
1.1	Web Service Description	2
1.2	Web Service Availability and Protocol.....	2
1.3	Web Service URLs.....	2
1.4	Web Service Examples	3
1.5	Web Service Exception Handling.....	4
1.6	Web Service Configurations.....	5
1.7	Security Requirements	5

Change History

Date	Changed by	Comments
Oct 15, 2007	Kammy Wesley	New document
October 16, 2007	Diane Klim	formatted and proofread
December 17, 2007	Yuxiao	Added date to query alerts; added exception handling; added how to configure it



1 Overview

The purpose of this document is to describe how to use the alert Web service.

1.1 Web Service Description

The alerts Web service returns data from the Alert table. The alert Web service allows for three different queries:

- Return all alerts for a team
- Return all alerts for an agent
- Return all alerts for all agents on all teams

Two constraints can be added to the above queries:

- from DateTime1
- to DateTime2

For each alert, the following information is returned:

- agent firstname
- agent lastname
- alert status (red or cleared)
- date and time the alert was issued
- rule text string for the alert
- text note saved for the alert

1.2 Web Service Availability and Protocol

The Web service is available in any deployed server of Frontline Advisor 2.8.4 or greater. The Web service is available via HTTP.

1.3 Web Service URLs

Three basic Web service URLs are available as follows:

- All Alerts:
`http://FA_HOST:FA_PORT/pa/ws/ALERTS`
- Team Alerts:
`http://FA_HOST:FA_PORT/pa/ws/ALERTS/TEAM/<teamid>`
- Agent Alerts:
`http://FA_HOST:FA_PORT/pa/ws/ALERTS/AGENT/<agentid>`

Note: ALERTS, TEAM, or AGENT are not case-sensitive.



Either **from**-dateTime or **to**-dateTime can be added to the above URLs as follows:

?from="dateTime1": which returns the alerts from this date time to current data time;

?to="dateTime2": which returns the alerts to this date time.

Both **from**-dateTime and **to**-dateTime can be added to the above URLs as follows:

?from="dateTime1"&to="dateTime2": which returns the alerts from the date time 1 to the date time 2.

The date-time formats of **from** and/or **to** can be as flexible as one of the following:

- YYYYMMDD
- YYYYMMDD HH:MM
- YYYYMMDD HH:MM:SS
- YYYY-MM-DD
- YYYY-MM-DD HH:MM
- YYYY-MM-DD HH:MM:SS
- YYYY/MM/DD
- YYYY/MM/DD HH:MM
- YYYY/MM/DD HH:MM:SS

Note: the date time formats of **from** and **to** do not need to be the same.

1.4 Web Service Examples

Example 1:

1. Get all alerts for all teams from 12:30, Dec. 17, 2007 to 13:00, Dec. 17, 2007.
[http://host:port/fa/ws/alerts?from="20071217 12:30"&to="20071217 13:00"](http://host:port/fa/ws/alerts?from=)

Returns:

```
<ALERTS>
<ALERT firstname="Kevi" lastname="Sherhouse" status="RED" datetime="2007-12-17 12:30:00.0" rule="Agent has no long calls." textnote="null" />
<ALERT firstname="Kevi" lastname="Sherhouse" status="RED" datetime="2007-12-17 12:30:00.0" rule="Agent has no calls under the avg wrap time." textnote="null" />
...
</ALERTS>
```



2. Get all alerts for team 9001 from 12:00 (midday), Dec. 17, 2007 to the current date and time.

`http://host:port/fa/ws/ALERTS/TEAM/9001?from="2007/12/17 12:00"`

Returns:

```
<ALERTS>
<ALERT firstname="Graz" lastname="Gasparini" status="RED" datetime="2007-12-17 12:30:00.0" rule="Agent has no long calls." textnote="null" />
<ALERT firstname="Graz" lastname="Gasparini" status="RED" datetime="2007-12-17 12:30:00.0" rule="Agent has no calls under the avg wrap time." textnote="null" />
<ALERT firstname="Graz" lastname="Gasparini" status="RED" datetime="2007-12-17 12:30:00.0" rule="Agent has no long wrap times." textnote="null" />
<ALERT firstname="Graz" lastname="Gasparini" status="RED" datetime="2007-12-17 12:30:00.0" rule="Agent is not putting any callers on hold." textnote="null" />
...
</ALERTS>
```

3. Get all alerts for agent 7028 until 13:00, Dec. 11, 2007.

`http://host:port/fa/ws/alerts/agent/7028?to="2007-12-11 13:00"`

Returns:

```
<ALERTS>
<ALERT firstname="Terry" lastname="Faridani" status="RED" datetime="2007-12-11 00:00:00.0" rule="Agent has no long calls." textnote="null" />
<ALERT firstname="Terry" lastname="Faridani" status="RED" datetime="2007-12-11 00:00:00.0" rule="Agent has no calls under the avg wrap time." textnote="null" />
<ALERT firstname="Terry" lastname="Faridani" status="RED" datetime="2007-12-11 00:00:00.0" rule="Agent has no long wrap times." textnote="null" />
<ALERT firstname="Terry" lastname="Faridani" status="RED" datetime="2007-12-11 00:00:00.0" rule="Agent is not putting any callers on hold." textnote="null" />
<ALERT firstname="Terry" lastname="Faridani" status="RED" datetime="2007-12-11 00:10:00.0" rule="Agent has no long calls." textnote="null" />
<ALERT firstname="Terry" lastname="Faridani" status="RED" datetime="2007-12-11 00:10:00.0" rule="Agent has no calls under the avg wrap time." textnote="null" />
<ALERT firstname="Terry" lastname="Faridani" status="RED" datetime="2007-12-11 00:10:00.0" rule="Agent has no long wrap times." textnote="null" />
...
</ALERTS>
```

1.5 Web Service Exception Handling

Exception 1: If this exception is caught while querying the total number of alerts to meet some constraints, the return of Web service is expected to be as follows.

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



```
<ALERTS failed="true" reason="Error: counting the total number of alerts to meet the conditions."/>
```

In this case, an exception should be generated in `fa_server.log` and may result from FA db access permission.

Exception 2: If the total number of alerts is too big, it may crush the FA server. So we constrain the number (how to configure it refers to next section). If this exception is caught, the return of Web service is expected to be as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<ALERTS row="166444" failed="true" reason="Warning: The number of rows is too big, please add or tighten constraints (team, agent, from or to)."/>
```

Note: 166444 is the total number of alerts to meet all constraints. In this case, the constraints should be tighter or added.

Exception 3: If this exception is caught while querying the alerts to meet some constraints, the return of Web service is expected to be as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<ALERTS failed="true" reason="Error: counting the total number of alerts to meet the conditions."/>
```

In this case, an exception should be generated in `fa_server.log`.

1.6 Web Service Configurations

The Web service can be configured under the following xml file:

```
{FA_deploy_dir}\Tomcat55\webapps\fa\WEB-INF\frontlineAdvisor-customerFacing-webServices.xml
```

The following properties can be configured:

- Maximum number of alerts allowed, search **maxRow** (default: 10000);
- Root name and child name of the return XML file, search **rootName** (default: ALERTS) or **childName** (default: ALERT);
- Description of exceptions, search **failedReason**;
- Return content type, if a validated xml doc is required, comment out **<property name="contentType" value="plain"/>**.

1.7 Security Requirements

No security/logins were created for the Web service.