



EMC® Watch4net®

Version 6.4u2

Frontend Administration Guide

P/N: pending

Rev : 50526

EMC²

Copyright ©2006 - 2014 EMC Corporation. All rights reserved. Published in the USA.

Published October, 2014

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (<https://support.emc.com>).

Contents

Preface	iii
Document Organization	iii
Where to get help	v
1 Tomcat Server Administration	1
1 Understanding the APG Web Portal	1
2 Tomcat Manager	2
2.1 APG Web Portal Configuration overview	2
3 Database Support and Naming Conventions	2
3.1 Database Installation	4
3.2 Indicating Datasources in Frontend Filters	5
3.3 Selecting Properties from Datasources in the Frontend	5
4 APG.xml	6
5 server.xml	9
6 Authentication and Single Sign-On	11
7 Multiple APG Web Portal instances	11
8 Search Engine Setup	12
9 Report Generation Time Tracking Setup	12
10 Tomcat Error Page	13
2 Scheduled Reports Administration	15
1 Overview	15
2 Installation	15
3 Frontend Configuration	16
3.1 Default configuration	16
3.2 Configuration content	16
4 Frontend Report Generator Configuration	17
4.1 Default configuration	17
4.2 Configuration content	27
5 Property-Store	28

- 6 Other configurations 29
- 7 Starting report scheduling 29
- 8 Formulas Management 30
- 9 Stored Reports 30
- 10 Troubleshooting 30
 - 10.1 Task Files 30
 - 10.2 Error Logging 31

Preface

Document Organization

1. Tomcat Server Administration	This section describes the web application administration in <i>Tomcat</i> .
---	--

Where to get help

EMC support, product, and licensing information can be obtained as follows:

PRODUCT INFORMATION - For documentation, release notes, software updates, or information about EMC products, go to EMC Online Support at:

<https://support.emc.com>

TECHNICAL SUPPORT - Go to EMC Online Support and click Service Center. You will see several options for contacting EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

1

Tomcat Server Administration

1 Understanding the *APG* Web Portal

Tomcat is a *Web application* server: it delivers dynamic web content based on some server-side application logic. The *APG* Web Portal is built on this concept, using some internal server-side logic to present on-the-fly generated reports.

The *APG* Web Portal is installed via the APG-Frontend module, and is contained in the frontend-`<version>-<OS>-x64.pkg` (64-bit), or frontend-`<version>-<OS>-x86.pkg` file (32-bit). The portal itself is a *Web application* which has to interact with the server. *APG*'s web apps, including the Frontend, are deployed automatically when Tomcat is first restarted after their installation.

2 Tomcat Manager

2.1 APG Web Portal Configuration overview

You may have to configure the *APG* Web Portal before starting it.

To do so, you need to configure two Tomcat files:

- *APG.xml*, located at `<APG>/Web-Servers/Tomcat/<instance name>/conf/Catalina/localhost`
This file declares all datasources to use as *ResourceLinks*, whose configurations are referenced in the *server.xml* file via the *global* attribute.

Note: *APG.xml* doesn't exist until Tomcat is started for the first time.

- *server.xml*, located at `<APG>/Web-Servers/Tomcat/<instance name>/conf`
This file contains the connection configurations for all source databases. Datasources declared in the *APG.xml* are linked to the databases declared here. This file is also used to configure the authentication, as well as for some other purposes.

Note: The syntax of the Tomcat files is documented fully in the Apache Tomcat documentation which you can refer to.

3 Database Support and Naming Conventions

Datasources are declared using *ResourceLink* elements in the *APG.xml* file, linking to a database in the *server.xml* file using *Resource* elements where their connection parameters reside.

Specifically, it is the value of the *global* attribute for the datasource in the *APG.xml* file that links it to a corresponding database entry in the *server.xml* file.

Datasources correspond to a pool of data which may have a one-to-one relationship with the contents of a database, or there can be several datasources contained in the same database. Datasource types, their characteristics, and the databases they are written to, is explained below.

When *APG* is configured to use several database, it will dispatch transparently requests to each of them, and show unified results in the resulting reports.

There are several types of available datasources, each with their own naming convention. And these are each written to specific databases.

Naming conventions must be strictly followed for the *ResourceLink* name attribute, which is the name of the datasource, in the *APG.xml* file. In the Frontend, when referencing a datasource, for example in the filter, it is the value of the *ResourceLink*'s name attribute that must be used. See [Indicating Datasources in Frontend Filters](#) on page 5.

Note: The Frontend User Interface refers to datasources as database names. This is for the convenience of the end-user. When making a selection, for example, what database to use to refresh topology data, you are in fact selecting the datasource.

First, there is always exactly one master datasource, which is always called `jdbc/master`, written to one master database. The master database contains user, profiles, roles, and rights data. For backward compatibility, if the `jdbc/master` database is not found, *APG-DB* will be used as the master database.

When referring to databases whose data is used to build reports, there are several types:

- Performance data is stored in timeseries databases and referenced with a timeseries datasource. This performance data is used to build reports. Performance data consists of a discrete sampling of performance indicators over a fixed period of time. A performance datasource must be named `jdbc/<datasource type>-<datasource name>`, where the datasource type is *APG*. For example, if you want to configure two timeseries datasources, they could be named `jdbc/APG-DB1` and `jdbc/APG-DB2`.

All performance datasources reference data in an instance of the *apg* database, discussed below in [server.xml](#) on page 9.

- Events databases, which are used to store event data, and are referenced with an events datasource. These are the databases, evidently, that are used to generate reports containing event data. An event datasource must be named `jdbc/<datasource type>-<datasource subtype>-name`, where the datasource type is *FLOW*. So, for example, the datasources that are pre-configured in *APG* to store event data from sources such as a Smarts-Listener, are called `jdbc/FLOW-RPE2-ARCH` and `jdbc/FLOW-RPE2-LIVE`. Another example is for VMWare, where its datasources can be called `jdbc/FLOW-VMWARE-EVENTS` and `jdbc/FLOW-VMWARE-TASKS`.

All (generic) events datasources are contained in an instance of the events database, discussed below in [server.xml](#) on page 9.

- There is a variation of the events datasource/database specifically for Traffic Flows (NetFlow, sFlow, derivations of NetFlow). It follows the same naming convention as other events datasources, `jdbc/<datasource type>-<datasource subtype>-name`, where the datasource type is FLOW, AND the subtype must be TF. All Traffic Flow datasources are contained in an instance of the `flow` database, discussed below in [server.xml](#) on page 9.

The following two database types are deprecated, replaced by the new event FLOW databases indicated above.

- RPE databases for ReportPack for Events data whose datasource must be named `jdbc/RPE-<datasource name>`. These databases are used in older implementations of the RPE, before APG version 5.0. In this case, all the modules that are integrated into the solution must correlate to this older version of APG.
- NetFlow databases for NetFlow-based data whose datasource must be named `jdbc/NF-<database name>`. These databases are used in older implementations of NetFlow, before APG version 5.1—both types of NetFlow database were available just for APG version 5.0. Similarly to the example above, all the modules that are integrated into the solution must correlate to this older version of APG.

If you want to make use of relationships between your infrastructure when building reports you must declare a topology database whose datasource must be named `topo/<datasource name>`. You could have multiple topology databases, for example, each corresponding to a performance database instance on a separate server, perhaps divided into topological groups like “core”, “mobile”, and “remote”, which you can make use of when building reports (see the APG *Frontend User Guide* for more information). The default topology datasource is named `topo/Topo-DB`, with each instance correlated to an instance of the `topology` database, discussed below in [server.xml](#) on 9.

If there are other database requirements for particular modules, these will be indicated in their own documentation.

3.1 Database Installation

Databases are automatically created by APG when you install particular modules as follows:

- By installing an instance of the **MySQL** module the following databases are created:

- master
- apg
- topology
- events

In each of the above cases, if you want additional instances of one particular type of database, you can install another instance of the **MySQL** module. You may only use one type of database on any particular instance, for example a second timeseries database on apg. The other types will be available in each instance but can remain unpopulated and idle.

- By installing an instance of the **Datastore** module the following databases are created:

- events
- flow

Similarly to above, in both cases, if you want additional instances of one particular type of database, you can install another instance of the **Datastore** module. Either one of the database types on any instance can remain unpopulated and idle.

3.2 Indicating Datasources in Frontend Filters

In the *APG* Frontend GUI, you can limit the data a filter selects by indicating its datasource of origin. For example, if you have two timeseries datasources named *APG-DB1* and *APG-DB2*, you can use the expression *#APG-DB1:ALL* to limit the data returned to the first datasource.

Additionally, if, for example, you have also have configured events databases, and want to limit the data returned to the two timeseries datasources indicated above, you can simply use the expression *#APG:ALL*—this would limit the data to the two *APG* timeseries datasources.

For more information on filters, see the *APG Frontend User Guide*.

3.3 Selecting Properties from Datasources in the Frontend

Users are able to select properties in the Frontend according to their datasource of origin.

This occurs in the Property Selection Helper dialog, where properties are divided by datasource, for example in the following manner:

- Properties from all timeseries databases are aggregated and appear under the *APG* tab. This is because all timeseries use the same schema.
- Properties from events databases appear under a tab corresponding to their datasource subtype, so for example, properties from *FLOW-VMWARE-EVENTS*, and *FLOW-VMWARE-TASKS* are aggregated together and appear under the *VMWARE* tab.
- Similarly all flow information will be aggregated under the *TF* tab.
- Depending on the source, properties in a particular tab may be an aggregation of one or more datasources. Other examples of tab headings include *RPE2*, *UCS*, and *SYSLOG*. For more information see individual module's documentation.

See the *APG Frontend User Guide* for more information on selecting properties in the Property Selection Helper dialog.

4 APG.xml

This *APG.xml* file serves several purpose:

- Declare all the datasources that this instance of the *APG* Frontend will access, which will correspond to the master database, as well as timeseries, events, flow, and topology databases. This declaration is made with a *ResourceLink* tag, where the *global* attribute is a reference to the database's connection parameters in the *server.xml* file.
- For the events datasources (including flows), the event mapping file that maps database properties to their Frontend equivalents is indicated for the *mapping* element. For more information, see the *APG-Event-Mapping Administration Guide*. As well, the table where events are written to in the database is indicated for the *table* element.

Refer to [Database Support and Naming Conventions](#) above for naming conventions.

The following is the default configuration for the *APG.xml* file.

Uncomment and/or add the required references to the databases you will use.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Context reloadable="false">
  <!-- The datasource definition are now located in conf/server.xml -->
  <!-- To use them in a web application, you must add a ResourceLink -->

  <!-- Gives access to the users, profiles, roles and rights. -->
  <ResourceLink name="jdbc/master" global="jdbc/master" type="javax.sql.DataSource" />

  <!-- Gives access to the timeseries database. -->
  <ResourceLink name="jdbc/APG-DB" global="jdbc/APG-DB" type="javax.sql.DataSource" />

  <!-- Gives access to the timeseries database with aligned aggregates. -->
  <!-- <ResourceLink name="jdbc/APG-DB:{ timezones=tz }" global="jdbc/APG-DB"
    type="javax.sql.DataSource" /> -->

  <!-- Gives access to the traffic flow data. -->
  <!--
  <ResourceLink name="jdbc/FLOW-TF-DATA:{ mapping=FLOW-TF-DATA.xml, table=raw_data }"
    global="jdbc/FLOW-TF-DATA" type="javax.sql.DataSource" />
  <ResourceLink name="jdbc/FLOW-TF-PROTO:{ mapping=FLOW-TF-PROTO.xml,
    table=aggr_protocols }" global="jdbc/FLOW-TF-DATA" type="javax.sql.DataSource" />
  <ResourceLink name="jdbc/FLOW-TF-APPLI:{ mapping=FLOW-TF-APPLI.xml,
    table=aggr_applications}" global="jdbc/FLOW-TF-DATA" type="javax.sql.DataSource" />
  -->

  <!-- Gives access to the events database. -->
  <!--
  <ResourceLink name="jdbc/FLOW-RPE2-ARCH:{ mapping=FLOW-RPE2.xml, table=events_archive }"
    global="jdbc/FLOW-RPE2" type="javax.sql.DataSource" />
  <ResourceLink name="jdbc/FLOW-RPE2-LIVE:{ mapping=FLOW-RPE2.xml, table=events_live }"
    global="jdbc/FLOW-RPE2" type="javax.sql.DataSource" /> -->

  <!-- Gives access to the topology database. -->
  <ResourceLink name="topo/Topo-DB" global="topo/Topo-DB" type="javax.sql.DataSource" />

  <!-- Data management tool needs to communicate with the timeseries backend. Please define
  one entry per available timeseries database, with the same name (ex: jdbc/APG-DB ->
  mgmt/APG-DB). -->
  <Resource name="mgmt/APG-DB" factory="org.apache.naming.factory.BeanFactory"
    type="com.watch4net.apg.v2.modules.admin.BackendCommunication" backendHost="localhost"
    backendTelnetInterfacePort="2001" backendSocketInterfacePort="2000" />
</Context>
```

```
</Context>
```

Also, follow the example at the end of the configuration file to add one data management tools entry per timeseries database.

VMWare Example

When you add new modules to the default *APG* installation/configuration, these new modules might need corresponding adjustments in the *APG.xml* file (as well as potentially in the *server.xml* file and other files).

For example, the following entry in the *APG.xml* file can be added to accommodate VMWare events and tasks:

```
<!-- VMWare Events and Tasks -->
<ResourceLink name="jdbc/FLOW-VMWARE-EVENTS:{_mapping=FLOW-VMWARE-EVENTS.xml,_table=
vmware_events_}" global="jdbc/FLOW-RPE2" type="javax.sql.DataSource" />
<ResourceLink name="jdbc/FLOW-VMWARE-TASKS:{_mapping=FLOW-VMWARE-TASKS.xml,_table=
vmware_tasks_}" global="jdbc/FLOW-RPE2" type="javax.sql.DataSource" />
```

As you can see, there are two *ResourceLink* entries added for VMWare, one for events and one for tasks. Two separate datasources are necessary because each of these types has their own schema. Whether the entry is for VMWare events, or tasks, is indicated in the name attribute (i.e. *jdbc/FLOW-VMWARE-EVENTS*), as well as in the mapping and table elements. In the case of *jdbc/FLOW-VMWARE-EVENTS*, it uses the mapping file *FLOW-VMWARE-EVENTS.xml* (all mapping files are located at *<APG>/Custom/WebApps-Resources/De* mapping), and writes to the table *vmware_events*.

Both of the VMWare tables are written to the *events* database corresponding to the global *jdbc/FLOW-RPE2* attribute value, as indicated for each entry. In this case, you just have to make sure the generic event database, *jdbc/FLOW-RPE2*, has been uncommented in the *server.xml* file.

In other cases, where there is a new database declared for a module in the global attribute, a new corresponding database entry will have to be added to the *server.xml* file. For example if you have two timeseries datasources, *APG-DB1* and *APG-DB2*, whose respective global attributes are *global="jdbc/APG-DB1"* and *global="jdbc/APG-DB1"*, database entries for each of these corresponding to the value of their global attributes would have to be configured in the *server.xml* file each pointing to a different *apg* timeseries database. Another example is for Broadsoft (*jdbc/FLOW-BROADSOFT-CDR*),

for instance, which would have to be correspondingly configured in the `server.xml` file. Once again, this will be covered in the individual module's documentation.

5 server.xml

This `server.xml` file serves several purposes:

- Declare all the databases this instance of the APG Frontend will access, and which are correspondingly referenced in the *APG.xml* file. The connection parameters for each database are indicated, and can be edited if necessary.
- Configure the authentication for access to the APG Web Portal. See [Authentication and Single Sign-On](#) below.
- Replace generic Tomcat error page. See [Tomcat Error Page](#) below.
- The other adjustments you make to the file are Tomcat-specific and are not covered in this guide. Refer to the Tomcat documentation for more information.

Refer to [Database Support and Naming Conventions](#) above for a general description of available databases.

Below are the database configurations that are enabled by default:

```
<!-- Defines the connection to the master database for MySQL. -->
<Resource name="jdbc/master" auth="Container" type="javax.sql.DataSource"
  maxActive="10" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
  testWhileIdle="true" validationQueryTimeout="5" timeBetweenEvictionRunsMillis=
  "10000" minEvictableIdleTimeMillis="60000" maxWait="10000" username="apg"
  password="watch4net" driverClassName="com.mysql.jdbc.Driver" removeAbandoned=
  "true" removeAbandonedTimeout="60" logAbandoned="true"
  url="jdbc:mysql://localhost:53306/master?autoReconnect=true" />

<!-- Defines the connection to the APG timeseries database for MySQL. -->
<Resource name="jdbc/APG-DB" auth="Container" type="javax.sql.DataSource" maxActive="30"
  maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false" testWhileIdle="true"
  validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
  minEvictableIdleTimeMillis="60000" maxWait="10000" username="apg"
  password="watch4net" driverClassName="com.mysql.jdbc.Driver" removeAbandoned="true"
  removeAbandonedTimeout="60" logAbandoned="true"
  url="jdbc:mysql://localhost:53306/apg?autoReconnect=true" />

<!-- Defines the connection to the topology database for MySQL. -->
```

```
<Resource name="topo/Topo-DB" auth="Container" type="javax.sql.DataSource" maxActive="30"
maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false" testWhileIdle="true"
validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
minEvictableIdleTimeMillis="60000" maxWait="10000" username="apg"
password="watch4net" driverClassName="com.mysql.jdbc.Driver" removeAbandoned="true"
removeAbandonedTimeout="60" logAbandoned="true"
url="jdbc:mysql://localhost:53306/topology?autoReconnect=true" />

<!-- Defines the connections to the APG Event database for MySQL. -->
<Resource name="jdbc/FL0W-RPE2" auth="Container" type="javax.sql.DataSource" maxActive="30"
maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false" testWhileIdle="true"
timeBetweenEvictionRunsMillis="10000" validationQueryTimeout="5"
minEvictableIdleTimeMillis="60000" maxWait="10000" username="apg"
password="watch4net" driverClassName="com.mysql.jdbc.Driver" removeAbandoned="true"
removeAbandonedTimeout="60" logAbandoned="true"
url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
/>
```

Comment or uncomment the databases entries in the file to match your overall *APG* configuration, which must be correlated to the configuration in the *APG.xml* file.

You may have to adjust the values of the connection attributes. For example for the *url* attribute, you may point to a different server where the database is being hosted, or change the connection port, for example, if there are multiple instances on the same machine connecting to different ports.

If you are adding an additional instance of a timeseries database, for example, you can duplicate the code for the second entry above (*Resource name="jdbc/APG-DB"*), edit its *Resource* name attribute to match the value of the *global* attribute indicated for the database in the *APG.xml* file, and adjust the *url* host and port according to the location of the second *apg* database.

You will find a lot of examples for multiple databases and multiple schemes in the default configuration file.

Note: The performance database declarations may slightly differ depending on which database engine you are using. You may have use multiple databases at the same time, using several engines.

6 Authentication and Single Sign-On

User authentication is achieved through the *Realm*. Most of the time, the *Realm* accesses a database to check user credentials. The *APG master* database should be used for this purpose unless you want to use an external authentication system.

Note: It is recommended to use the *master* database for authentication. If you are using an external authentication, it is your responsibility to ensure consistency between your user databases. Each authenticated user must have a profile in the *master* database.

You can have one *Realm* per application but you can also declare a single *Realm* for all your application. This way, you can use single sign-on for all *APG* application.

The single sign-on is enabled by default, that's why you will not see any *Realm* in the default *APG* configuration file. The common *Realm* is declared in `conf/server.xml`.

```
<!-- Single sign on based on APG master database -->
<Realm className="org.apache.catalina.realm.JDBCRealm"
  driverName="com.mysql.jdbc.Driver"
  connectionURL="jdbc:mysql://localhost:53306/master?autoReconnect=true"
  connectionName="apg" connectionPassword="watch4net"
  userTable="users" userNameCol="username" userCredCol="password"
  userRoleTable="users" roleNameCol="rolename" />
<Valve className="org.apache.catalina.authenticator.SingleSignOn"/>
```

WARNING: Make sure that the database declared here is **ALWAYS** in sync with your *master* database. Otherwise you could get unexpected results.

7 Multiple APG Web Portal instances

It is sometimes useful to have several portal instances to manage several databases. You can do this by installing additional *APG-Frontend* instances with the Module Manager. See the *APG Module Manager Guide* for more information.

8 Search Engine Setup

Before being used, the search engine has to be configured **and initialized**. Fortunately, it uses the same configuration for both the generation process and the reading process.

By default, the configuration file in use is the one located in the Frontend-Search default installation directory: `Tools/Frontend-Search/Default/conf/frontend-search.xml`. The location of this configuration file can be easily changed by editing the `common.properties` file of the frontend and setting the `apg.frontend.search.config` key with the path to the desired configuration file.

Note: This path can be either absolute either relative to the installation directory of the Tomcat server: for instance `<APG>/Web-Servers/Tomcat/Default`.

Once the configuration has been set, **the search index needs to be generated once before being available**. Please see the documentation that comes with the `frontend-search` module in order to get detailed information about this process.

Note: Frontend search generation may have been scheduled with its default configuration during installation.

9 Report Generation Time Tracking Setup

Tracking the generation time of a troubling report in detail is possible with the APG tracking tools. The default configuration of the tracking engine is available in the file named `Web-Applications/APG-Frontend/APG/conf/com` with the parameters starting by `apg.tracking`. Some properties can also be adjusted on the fly with a JMX call like the time threshold for example. It is suggested to use JConsole and work with the object

```
com.watch4net.apg.tracker:instance=/APG
```

In the file mentioned above, you will be able to modify the following tracking properties:

- **apg.tracking.export.threshold.default** The default threshold for the XML export. A generation time report will only be exported if the total time in milliseconds exceed the threshold. The default value, if not defined, is `-1` which disable the export of reports while a value of `0` will export every task.

- **apg.tracking.export.max.count** The number of reports to export per task name. Default value if not defined is 5.
- **apg.tracking.output.directory** The directory containing the exported reports currently in Web-Server/Tomcat/instance/logs. Default value if not defined is ./

10 Tomcat Error Page

To avoid displaying sensitive server information such as version number when webapps crash, a specific page has been implemented. This latter only displays HTTP status code and a short description of the current error.

By default, this feature is activated. But if you updated Tomcat from a version lesser or equals to 6.4, you have to manually turn on this feature. To do so, in the `server.xml` file add the XML attribute `errorReportValveClass` to the Host tag and set it to `"com.watch4net.apg.v2.gui.tomcat.W4NErrorReportValve"`:

```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="false"
      xmlValidation="false" xmlNamespaceAware="false"
      errorReportValveClass="com.watch4net.apg.v2.gui.tomcat.W4NErrorReportValve">
```


2

Scheduled Reports Administration

1 Overview

To reduce the load of the APG's web server, Scheduled Reports are scheduled and generated by separate modules: the Task Scheduler and the Frontend Report Generator. The two modules may be either deployed locally along with the Frontend or remotely on a scheduling server dedicated to report scheduling and generation.

2 Installation

The installation consists in installing a Task Scheduler and a Frontend Report Generator modules on the APG server that will be used as scheduling server.

The installation also requires the Webservice Gateway service to be installed along with the above modules.

For more details on how to install modules and start services, please refer to the **Module Manager Guide**.

WARNING: All three modules must be deployed on the same APG installation.

3 Frontend Configuration

Once the modules are installed, APG Frontend must be configured to access the scheduling server.

The configuration file is located in

<APG>/Custom/WebApps-Resources/<instance>/scheduling/scheduling-servers.xml, where <APG> is the root directory of the APG installation where the Frontend is deployed.

The configuration file mainly contains the settings used to connect to the Webservice Gateway. For more information on the Webservice Gateway, please refer to the *APG Webservice Gateway Admin Guide*.

3.1 Default configuration

The default configuration contains the settings for an installation where the Frontend and the three scheduling modules are located on the same APG server. It also assumes that the instance name of the Task Scheduler and Frontend Report Generator modules are Default.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<scheduling-servers xmlns="http://www.watch4net.com/APG/Frontend/Scheduled-Reports/Scheduling-
Servers">
  <server validateSSL="false" password="changeme" username="admin" url="https://
localhost:48443/" />
</scheduling-servers>
```

3.2 Configuration content

- <server> Each <server> tag defines the location of a scheduling server. Currently, only one server is supported.

The server attributes are:

- `url` (required) URL of the Webservice Gateway on the scheduling server.
- `username` (optional) user name used for the connection. Default: no user name.
- `password` (optional) authentication password used for the connection. Might be encrypted. Default: no password.
- `validateSSL` (optional) enable or disable SSL validation for the connection. Default: false.
- `task-scheduler-instance` (optional) instance name of the Task Scheduler on the scheduling server. Default instance: Default.
- `report-generator-instance` (optional) instance name of the Frontend Report Generator on the scheduling server. Default instance: Default.

Note: If no server tag is specified, report scheduling capabilities of APG are disabled, including scheduled and pre-generated reports.

4 Frontend Report Generator Configuration

The Frontend Report Generator might need to be configured to access the APG databases. Its configuration file is `report-generation-config.xml`, located in the `<APG Root>/Tools/Frontend-Report-Generator/<Instance>/conf` directory of the Scheduling server. It contains the settings required to access the APG's Master and Timeseries databases and is validated by the XML Schema `report-generation-config.xsd`. Basically, it should contain the same database settings as the [server.xml](#) file (described on 9).

This section covers only the basics of the Frontend Report Generator's configuration. For more advanced details, please refer to the *APG Frontend Report Generator Admin Guide*.

4.1 Default configuration

The default configuration contains the settings for a Frontend Report Generator connected to a single MySQL database located on the same server.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns="http://www.watch4net.com/APG/Frontend/Report-Generation/Config"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.watch4net.com/APG/Frontend/Report-Generation/Config_report-
generation-config.xsd">

  <master>
    <!-- Defines the connection to the master database for MySQL. -->
    <datasource driverClassName="com.mysql.jdbc.Driver"
      url="jdbc:mysql://localhost:53306/master?autoReconnect=true"
      username="apg" password="watch4net"
      maxActive="10" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
      timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
      removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" />

    <!-- Example of connection to the master database for Oracle 10g. -->
    <!--
    <datasource driverClassName="oracle.jdbc.OracleDriver"
      url="jdbc:oracle:thin:@localhost:1521:APG"
      username="apg" password="watch4net"
      maxActive="10" maxIdle="5" validationQuery="SELECT 1 FROM DUAL" testOnBorrow="false"
testWhileIdle="true"
      timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
      removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
      connectionProperties="SetBigStringTryClob=true" />
    -->
  </master>

  <database>
    <!-- Defines the connection to the APG timeseries database for MySQL. -->
    <datasource id="APG-DB:{_cachegrp=DB_}" driverClassName="com.mysql.jdbc.Driver"
      url="jdbc:mysql://localhost:53306/apg?autoReconnect=true"
      username="apg" password="watch4net"
      maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
      timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
      removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>

    <!-- Example of connection to the APG timeseries database for Oracle 10g. -->
    <!--

```

```

<datasource id="APG-DB:{ cachegrp=DB }" driverClassName="oracle.jdbc.OracleDriver"
  url="jdbc:oracle:thin:@localhost:1521:APG"
  username="apg" password="watch4net"
  maxActive="30" maxIdle="5" validationQuery="SELECT 1 FROM DUAL" testOnBorrow="false"
  testWhileIdle="true"
  timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
  removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" />
-->

<!-- Example of connection to the APG Traffic Flow data. -->
<!--
<datasource id="FLOW-TF-DATA:{ mapping=FLOW-TF-DATA.xml, table=FLOW_RAW_DATA__SEQ_0_4__
}"
  maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
  testWhileIdle="true"
  validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
  minEvictableIdleTimeMillis="60000"
  maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
  removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
  url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
<datasource id="FLOW-TF-PROTO:{ mapping=FLOW-TF-PROTO.xml, table=
FLOW_PROTOCOL__SEQ_0_4__}"
  maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
  testWhileIdle="true"
  validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
  minEvictableIdleTimeMillis="60000"
  maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
  removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
  url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
<datasource id="FLOW-TF-APP:{ mapping=FLOW-TF-APP.xml, table=
FLOW_APPLICATION_TOPN__SEQ_0_4__}"
  maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
  testWhileIdle="true"
  validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
  minEvictableIdleTimeMillis="60000"
  maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
  removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
  url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />

```

```

        <datasource id="FLOW-TF-SRCADDR:{ mapping=FLOW-TF-SRCADDR.xml, table=
FLOW_SRCADDR_TOPN__SEQ_0_4__}"
            maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
            validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
minEvictableIdleTimeMillis="60000"
            maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
            url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
        <datasource id="FLOW-TF-DSTADDR:{ mapping=FLOW-TF-DSTADDR.xml, table=
FLOW_DSTADDR_TOPN__SEQ_0_4__}"
            maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
            validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
minEvictableIdleTimeMillis="60000"
            maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
            url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
        <datasource id="FLOW-TF-CONV:{ mapping=FLOW-TF-CONV.xml, table=
FLOW_CONVERSATION_TOPN__SEQ_0_4__}"
            maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
            validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
minEvictableIdleTimeMillis="60000"
            maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
            url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
        <datasource id="FLOW-TF-HOSTS:{ mapping=FLOW-TF-HOSTS.xml, table=FLOW_HOSTS__SEQ_0_4__
}"
            maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
            validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
minEvictableIdleTimeMillis="60000"
            maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
            url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
        <datasource id="FLOW-TF-NOSNMP:{ mapping=FLOW-TF-NOSNMP.xml, table=
FLOW_NOSNMP__SEQ_0_4__}"

```

```

        maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
minEvictableIdleTimeMillis="60000"
        maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
        url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
    <datasource id="FLOW-TF-ALLPORTS:{ mapping=FLOW-TF-ALLPORTS.xml, table=
FLOW_ALLPORTS__SEQ_0_4__}"
        maxActive="10" maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        validationQueryTimeout="5" timeBetweenEvictionRunsMillis="10000"
minEvictableIdleTimeMillis="60000"
        maxWait="10000" username="flow" password="watch4net" driverClassName="com.watch4net.
x100.jdbc.Driver"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
        url="jdbc:x100://localhost:48443/Databases/APG-Datastore" />
    -->

    <!-- Defines the connections to the APG Event database for MySQL. -->
    <!--
    <datasource id="FLOW-RPE2-ARCH:{ mapping=FLOW-RPE2.xml, table=events_archive }"
driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
    <datasource id="FLOW-RPE2-LIVE:{ mapping=FLOW-RPE2.xml, table=events_live }"
driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
    -->

```

```

        <!-- Defines the connections to the APG Event database for APG Datastore. -->
        <!--
        <datasource id="FLOW-RPE2-ARCH:{ mapping=FLOW-RPE2.xml, table=events_archive }"
            url="jdbc:x100://localhost:48443/Databases/APG-Datastore/Default" maxActive="10"
            maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false" testWhileIdle="true"
            timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
        = "10000"
            username="admin" password="changeme" driverClassName="com.watch4net.x100.jdbc.Driver
        "
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" />
        <datasource id="FLOW-RPE2-LIVE:{ mapping=FLOW-RPE2.xml, table=events_live }"
            url="jdbc:x100://localhost:48443/Databases/APG-Datastore/Default" maxActive="10"
            maxIdle="10" validationQuery="SELECT 1" testOnBorrow="false" testWhileIdle="true"
            timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
        = "10000"
            username="admin" password="changeme" driverClassName="com.watch4net.x100.jdbc.Driver
        "
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" />
        -->

        <!-- Defines the connections to the emc-vnx events database for MySQL. -->
        <datasource id="FLOW-VNX-LIVE:{ mapping=FLOW-VNX.xml, table=nasusers }" driverClassName
        = "com.mysql.jdbc.Driver"
            url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
            username="apg" password="watch4net"
            maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
        testWhileIdle="true"
            timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
        10000"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>

        <!-- Defines the connections to the emc-vipr events database for MySQL. -->
        <datasource id="FLOW-VIPR-EVENTS:{ mapping=FLOW-VIPR.xml, table=vipr_events }"
        driverClassName="com.mysql.jdbc.Driver"
            url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
            username="apg" password="watch4net"
            maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
        testWhileIdle="true"
            timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
        10000"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>

```

```

    <!-- Defines the connections to the vmware events database for MySQL. -->
    <datasource id="FLOW-VMWARE-EVENTS:{ mapping=FLOW-VMWARE-EVENTS.xml, table=
vmware_events_}" driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
    <datasource id="FLOW-VMWARE-TASKS:{ mapping=FLOW-VMWARE-TASKS.xml, table=vmware_tasks_}"
" driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>

    <!-- Defines the connections to the cisco-ucs-events database for MySQL. -->
    <!--
    <datasource id="FLOW-UCS-LIVE:{ mapping=FLOW-UCS.xml, table=ucs_live}" driverClassName
="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
    -->

    <!-- Defines the connections to the emc-prosphere events database for MySQL. -->
    <!--
    <datasource id="FLOW-PROSPHERE-ARCH:{ mapping=FLOW-UEVENTS.xml, table=
unifiedevents_archive }" driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait

```

```

="10000"
    removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
    <datasource id="FLOW-PROSPHERE-LIVE:{ mapping=FLOW-UEVENTS.xml, table=
unifiedevents_live }" driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
    removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
-->

<!-- Defines the connections to the emc-som-events database for MySQL. -->
<!--
    <datasource id="FLOW-SOM-ARCH:{ mapping=FLOW-EMC-SOM.xml, table=emcsom_archive }"
driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
    removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
    <datasource id="FLOW-SOM-LIVE:{ mapping=FLOW-EMC-SOM.xml, table=emcsom_live }"
driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
="10000"
    removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
-->

<!-- Defines the connections to the what-if scenarios database for MySQL. -->
<datasource id="FLOW-WHATIF-SCENARIOS:{ mapping=FLOW-WHATIF-SCENARIOS.xml, table=
whatif_scenarios}" driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1" testOnBorrow="false"
testWhileIdle="true"

```

```

timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>

<!-- Defines the connections to the generic event schema for MySQL. -->
<datasource id="FLOW-EVENTS-GENERIC:{_mapping=FLOW-GENERIC-EVENT.xml,_table=
genericevents_live_}" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
username="apg" password="watch4net"
maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
<datasource id="FLOW-EVENTS-GENERICARCH:{_mapping=FLOW-GENERIC-EVENT.xml,_table=
genericevents_archive_}" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:53306/events?autoReconnect=true&tinyInt1isBit=false"
username="apg" password="watch4net"
maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>

<!-- Defines the connections to the APG Compliance database for MySQL. -->
<datasource id="FLOW-COMPLIANCE-BREACH:{_mapping=FLOW-COMPLIANCE-BREACH.xml,_table=
breach_}" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:53306/compliance?autoReconnect=true&tinyInt1isBit=
false"
username="apg" password="watch4net"
maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
<datasource id="FLOW-COMPLIANCE-POLICY:{_mapping=FLOW-COMPLIANCE-POLICY.xml,_table=
policy_}" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:53306/compliance?autoReconnect=true&tinyInt1isBit=
false"
username="apg" password="watch4net"
maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"

```

```

        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
        <datasource id="FLOW-COMPLIANCE-RULE:{_mapping=FLOW-COMPLIANCE-RULE.xml,_table=rule_}"
driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/compliance?autoReconnect=true&tinyInt1isBit=
false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
        <datasource id="FLOW-COMPLIANCE-CONFIGCHANGE:{_mapping=FLOW-COMPLIANCE-CONFIGCHANGE.xml
,_table=config_change_}" driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:53306/compliance?autoReconnect=true&tinyInt1isBit=
false"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"/>
    </database>

    <topology>
        <!-- Defines the connection to the topology database for MySQL. -->
        <datasource id="Topo-DB" driverClassName="com.mysql.jdbc.Driver"
            url="jdbc:mysql://localhost:53306/topology?autoReconnect=true"
            username="apg" password="watch4net"
            maxActive="30" maxIdle="5" validationQuery="SELECT_1" testOnBorrow="false"
testWhileIdle="true"
            timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait="
10000"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
connectionProperties="cachePrepStmts=true;useServerPrepStmts=false;maintainTimeStats=false;
dontTrackOpenResources=true;enableQueryTimeouts=false;processEscapeCodesForPrepStmts=false;
jdbcCompliantTruncation=false;rewriteBatchedStatements=true;" />

        <!-- Example of connection to the topology database for Oracle 10g. -->
        <!--
        <datasource id="Topo-DB" driverClassName="oracle.jdbc.OracleDriver"

```

```

        url="jdbc:oracle:thin:@localhost:1521:APG"
        username="apg" password="watch4net"
        maxActive="30" maxIdle="5" validationQuery="SELECT 1 FROM DUAL" testOnBorrow="false"
        testWhileIdle="true"
        timeBetweenEvictionRunsMillis="10000" minEvictableIdleTimeMillis="60000" maxWait
        ="10000"
        removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
        connectionProperties="SetBigStringTryClob=true" />
    -->
</topology>

<alerting>
    <!-- Alerting Properties -->
    <manager
        name="Local_Manager"
        user="admin"
        pass="changeme"
        url="service:jmx:rmi:///jndi/rmi://localhost:52569/jmxrmi" />
    </alerting>
</configuration>

```

4.2 Configuration content

- <master> Contains the settings of the Master datasource used for report generation.
 - <datasource driverClassName="*driverClassName*" url="*url*" username="*username*" password="*password*">: the master datasource used by the APG Frontend.
 - * *driverClassName* (required): the fully qualified Java class name of the JDBC driver to be used.
 - * *url* (required): the JDBC connection URL.
 - * *username* (required): the user name used to connect to the database.
 - * *password* (required): the password used to connect to the database.
- <database> Contains the connection parameters for the timeserie, event and flow databases used for report generation.
 - <datasource id="*id*" driverClassName="*driverClassName*" url="*url*" username="*username*" password="*password*">: a datasource used by the APG

- * *id* (required): the APG datasource ID (<database type>-<database name>, e.g. APG-DB).
 - * *driverClassName* (required): the fully qualified Java class name of the JDBC driver to be used.
 - * *url* (required): the JDBC connection URL.
 - * *username* (required): the user name used to connect to the database.
 - * *password* (required): the password used to connect to the database.
- <topology> Contains the connection parameters of the topology databases used for report generation. As for the <database> tag, topology datasources are given by a <datasource> tag.
 - <alerting> Contains the connection parameters of the alerting managers that may be configured as recipients for generated reports.
 - <manager name="*name*" user="*user*" pass="*password*" url="*url*">
: an alerting manager used by APG
 - * *name* (required): the name of the alerting manager.
 - * *user* (required): the user name used to connect to the alerting manager.
 - * *pass* (required): the password used to connect to the alerting manager.
 - * *url* (required): the URL of the alerting manager.

5 Property-Store

Similarly to the Frontend and other web-applications, report generation may be speed-up in the Frontend Report Generator by taking advantage of the Property-Store module.

The setup of the Property-Store requires the following steps:

- **Property-Store module installation**
During the installation, make sure to enable the `import-properties.task` file when prompted.
- **Datasources configuration update**
In the configuration file (`report-generation-config.xml`), a `cachegrp` attribute must be specified for each Database which will use the Property-Store.
- **import-properties.task configuration**
The Property-Store scheduled a task for periodical import of properties. The file is located in the directory
`<APG>/Tools/Task-Scheduler/<instance>/data/task_repository/property-store/Default/.`

By default the task is configured to work with the Web-Server. To configure it for the Frontend Report Generator only, replace the lines:

```
<parameters>
  <param>--server</param>
  <param>../../../../Web-Servers/Tomcat/Default/conf/server.xml</param>
  <param>--application</param>
  <param>../../../../Web-Servers/Tomcat/Default/conf/Catalina/localhost/APG.xml</param>
</parameters>
```

by (if Default is the Frontend-Report-Generator instance to use):

```
<parameters>
  <param>--application</param>
  <param>../../../../Tools/Frontend-Report-Generator/Default/conf/report-generation-config.xml</param>
</parameters>
```

- **Start the Task Scheduler**
- **Run the import-properties task** (optional)

The task is scheduled to run every hour. To get the properties imported immediately, the task may be executed manually using the Centralized Management, under the Database category.

For more details on the Property-Store installation and the configuration of APG datasources, please refer to the *Property-Store Guide*.

6 Other configurations

Normally, once the Frontend and Frontend Report Generator modules are correctly configured, the Report Scheduling server can be started. In some specific cases though, the Task Scheduler's and the Webservice Gateway's configurations might also need to be modified. If needed, please refer to their respective documentations for more details.

7 Starting report scheduling

Once the Scheduling server is installed and configured, the last step is to start the Task Scheduler and Webservice Gateway services. Afterwards, Scheduled reports can be scheduled and modified from the Web Portal.

8 Formulas Management

Formulas used during report generation are located in the Formula Resources module of the APG installation. Consequently, if a remote scheduling server is used, several Formulas Resources modules are needed: one for the Frontend Report Generator installation, and one for each Web Portal.

In such case, all the Formulas Resources modules must always stay synchronized between each other. Failing to synchronize will lead to unexpected differences between real-time and scheduled reports.

WARNING: Every addition, removal or change of a formula must be applied to each Frontend Report Generator module.

9 Stored Reports

The Frontend Report Generator stores the reports it generates locally on the scheduling server. To be available on the Frontend, the stored reports are periodically synchronized between the scheduling server and the Frontend.

By default the synchronization is performed every hour. This period can be modified by setting the Java system property `remote.updatable.report.serializer.update.period` to a given number of seconds.

10 Troubleshooting

10.1 Task Files

Technically, the scheduling of report is entirely managed by the Task Scheduler. For each scheduled report, a task file is created in the Task Scheduler. These task files contains not only the scheduling (a cron expression) of the report, but also all the informations needed for the report generation.

Consequently, it is also possible to manage scheduled reports through the Centralized Management, in the Task section. Tasks files relative to report scheduling can be found under the category `report-scheduling`.

WARNING: Task files are in XML format and are human readable. It is however highly discouraged to modify the

task files directly. In case of error, the corresponding report might not be generated anymore or without the appropriate options.

10.2 Error Logging

Most of report scheduling and generation errors can be retrieved from the APG Web Portal. Details and statistics about report scheduling can be found in the scheduling part of the admin section. For more information on the Web Portal, please refer to the *APG Frontend User Guide*.

More details information about scheduled reports generation might also be obtained directly from the Task Scheduler module. Each time a report is generated, a log file containing details about the generation is created.

The log files are located inside the Task Scheduler, in the directory `logs/scheduled-reports/`.

Note: Retrieving the log file of a specific scheduled report directly from the file system is difficult. When available, it is advised to access them by the Centralized Management.
