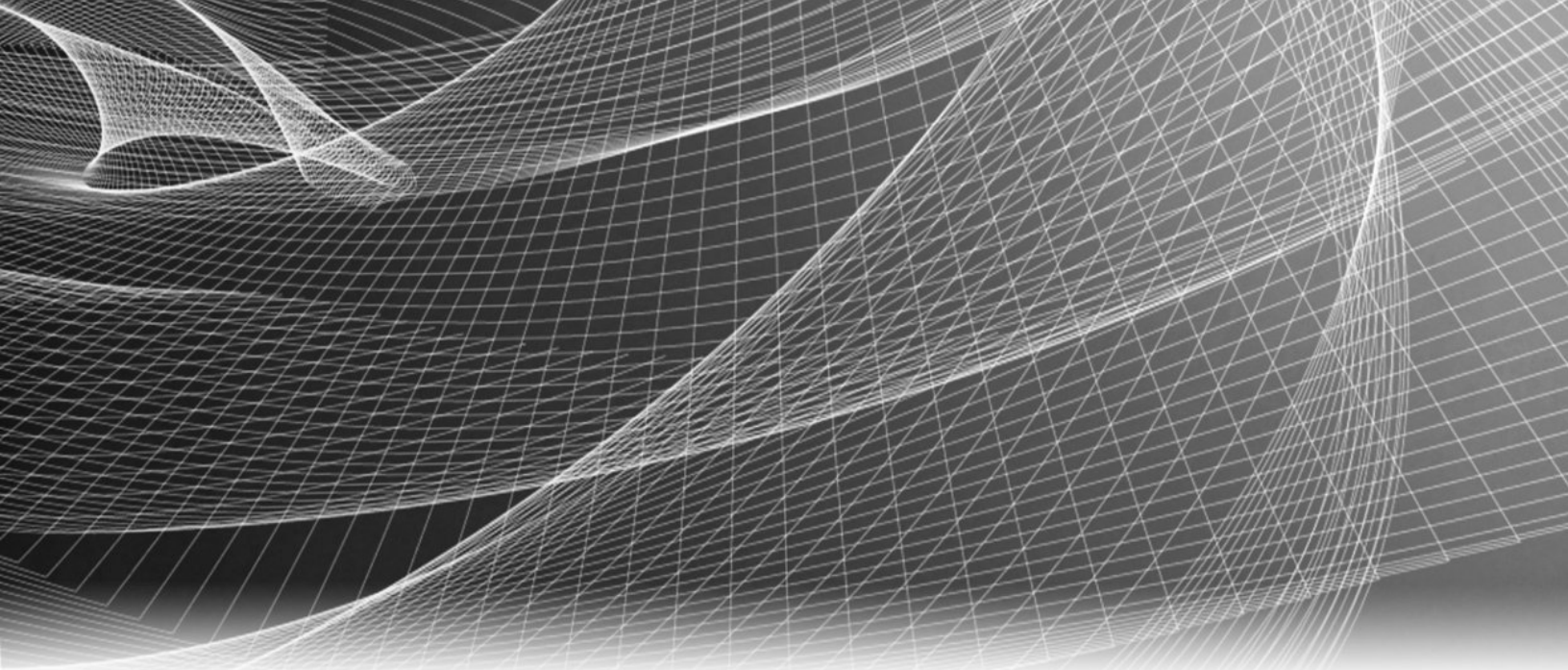




EMC[®] Watch4net[®]

Version 6.4u2

Property Store Administration Guide

P/N: pending

Rev : 40030

Copyright ©2010 - 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
Purpose	iii
Intended Audience	iii
Prerequisites	iii
Document Organization	iii
Where to get help	v
1 Architectural Overview	1
2 Configuration	3
1 Import Properties task	3
2 Web Applications configuration	3
3 Other Modules configuration	4

Preface

Purpose

The APG *Property Store* is a dedicated storage engine that is able to cache locally the APG metrics properties to speed-up the properties related operations (filter resolution, grouping, ...) and avoid unnecessary queries to the databases. It is typically installed where report generation or more generally database queries take place.

Intended Audience

This guide is addressed to the APG system administrator that wants to configure the APG *Property Store*.

Prerequisites

The APG *Property Store* assumes that the APG *Task Scheduler* is installed on the same machine to periodically refresh the cached properties. The default *import-properties* task assumes that *Tomcat* and the APG *Web Portal* are installed with their default instance.

Document Organization

1. Architectural Overview	This section describes the APG <i>Property Store</i> and its basic functionality.
2. Configuration	This section explains how to configure the APG <i>Property Store</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.

Architectural Overview

The *APG Property Store* imports properties from various *APG* metric databases distributed across the network. These databases can be grouped together to form local *cache groups*.

Each *APG* database is periodically imported individually and then merged into their respective *cache groups*. A database is imported only if it has changed and a group is merged only when at least one database has been imported.

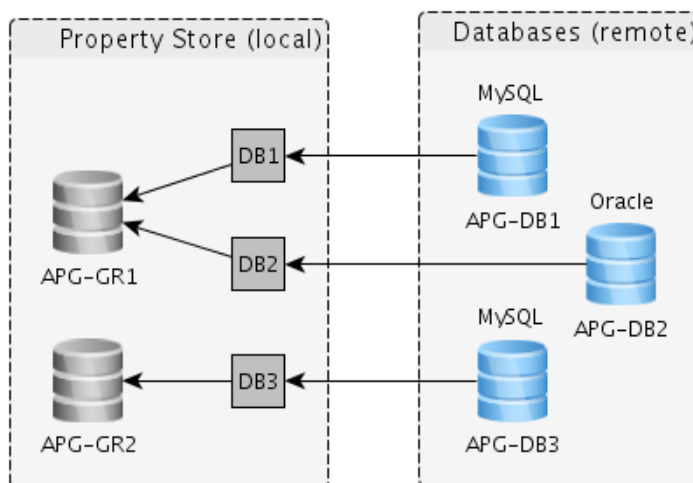


Figure 1: Property Store Import

In the previous example, we have defined 2 groups, linked to respectively two and one remote *APG* database.

Once imported, the *APG Property Store* can be used by the *APG* database accessor layer to speed-up property access (filtering, expansion, ...) and avoid unnecessary

queries to the remote databases.



Before the initial import, no data will be available in the reports. When using the APG *Property Store*, the APG database accessor will never fallback to individual databases.

Starting at APG 6.1, the APG *Property Store* is fully supported by the following modules:

- Every web application that access an APG database (APG *Web Portal*, APG *Web Service*, ...)
- The APG *Frontend Search* CLI
- The APG *Frontend Report Generator*



When multiple modules access the same APG *Property Store*, make sure that all their configuration files are consistent (i.e. they are connected to the same databases using the same names, and that they are grouped the same way).

2

Configuration

1 Import Properties task

The APG *Property Store* import properties periodically by scheduling an *import-properties* task into the APG *Task Scheduler*.

The *import-properties* task supports various configuration files format (`server.xml` / `APG.xml`, `frontend-search.xml`, ...) to get the list of APG databases.

```
<?xml version="1.0" encoding="UTF-8"?>
<task name="import-properties">
  <!-- task running every 6 hours starting at 1h00 each day. -->
  <schedule xsi:type="schedule-repeated" cron="0 1,7,13,19 * * * ?"></schedule>
  <!-- ... -->
  <parameters>
    <!-- Tomcat server configuration (optional, applies to webapps only). -->
    <param>--server</param>
    <param>path/to/Tomcat/Default/conf/server.xml</param>
    <!-- Application configuration. -->
    <param>--application</param>
    <param>path/to/Tomcat/conf/Catalina/localhost/APG.xml</param>
  </parameters>
  <!-- ... -->
</task>
```

The previous example schedules an import every 6 hour and uses APG *Web Portal* configuration file. This is the default setup.

2 Web Applications configuration

In order to be used with the APG *Property Store*, databases declared in `APG.xml` (or any similar web-application configuration file, for example `APG-WS.xml`) have to specify a *cachegrp* attribute.

This attribute allows the *import-properties* task to know in which group the database should be merged and the web application to determine which group it should redirect

properties related queries to.

If a database doesn't have a *cachegrp*, it won't get imported and the web application won't use the APG *Property Store* to speed-up properties related queries.

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

    <!-- ... -->

    <!-- Import APG-DB1 properties into the group GRP1. -->
    <ResourceLink name="jdbc/APG-DB1:{_cachegrp=GRP1_}" ... />

    <!-- Import APG-DB2 properties into the group GRP1. -->
    <ResourceLink name="jdbc/APG-DB2:{_cachegrp=GRP1_}" ... />

    <!-- Import APG-DB3 properties into the group GRP2. -->
    <ResourceLink name="jdbc/APG-DB3:{_cachegrp=GRP2_}" ... />

    <!-- Dont import APG-DBX properties into the Property Store. -->
    <ResourceLink name="jdbc/APG-DBX" ... />

    <!-- ... -->

</Context>
```



When timezone alignment is used (i.e. when the attribute *timezones* is set in database declaration), every database in the same *cachegrp* must share the same *timezones* setting.

3 Other Modules configuration

Similarly, other supported modules which declare *APG* datasources have to specify the *cachegrp* attribute.

Their configuration files can also be used by the *import-properties* to define which database should be imported.

If a database doesn't have a *cachegrp*, it won't be imported and the application won't use the APG *Property Store* to speed-up properties related queries.

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

    <!-- ... --->

    <index-configuration>

        <!-- ... --->

        <apg>
            <!-- Import APG-DB1 properties into the group GRP1. -->
            <datasource id="jdbc/APG-DB1:{_cachegrp=GRP1_}" ... />

            <!-- Import APG-DB2 properties into the group GRP1. -->
            <datasource id="jdbc/APG-DB2:{_cachegrp=GRP1_}" ... />

        </apg>

    </index-configuration>

</configuration>
```

```
<!-- Import APG-DB3 properties into the group GRP2. -->
<datasource id="jdbc/APG-DB3:{_cachegrp=GRP2_}" ... />

<!-- Dont import APG-DBX properties into the Property Store. -->
<datasource id="jdbc/APG-DBX" ... />
</apg>

</index-configuration>

<!-- ... --->

</configuration>
```

The previous example shows the APG *Frontend Search* CLI configuration file. The APG *Frontend Report Generator* would use similar syntax.