



EMC® Watch4net®

Version 6.4u2

Frontend Search Administration Guide

P/N: pending

Rev : 41427

EMC²

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
Where to get help	v
1 Product Overview	1
2 Architectural Overview	3
3 Installation	7
4 Configuration	9
1 Index configuration	10
1.1 Indexed values key	12
2 Search configuration	13
3 Split indexed values and search keywords	14
5 Start Up	15
6 Filter nodes in the search index	17
1 Include specific root nodes	18
2 Exclude specific nodes	18
7 Schedule search index generation	19
1 Use the module scheduler	20
2 Use the operating system task scheduler	20
3 Use the APG task scheduler	20
4 Schedule multiple configurations for different days of the week	21
8 Advanced search	25

Preface

Purpose

This guide provide configuration information about the Frontend Search module.

Intended Audience

This guide is intended to be read by any *APG* administrator who will configure the Frontend Search module to enable search in the APG Frontend.

Prerequisites

It is assumed that the *APG* application has been properly installed and configured on the server. The `Backend Adapter` license is required to run the Frontend Search module.

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.

Product Overview

The Frontend Search module allows the APG Frontend to provide a full search experience. The Frontend Search indexes the APG Frontend tree reports and it is used by the APG Frontend to find the search results.

Here are the requirements:

- APG Frontend 5.1 (any update level)
- the Backend Adapter license

2

Architectural Overview

The Frontend Search module is used by the APG Frontend module. The Frontend Search module is used to generate the search index. The search index is a directory that must be located on the same server as the APG Frontend.

It is recommended to use a `data_property_flat` table to improve performance.

Example of a single server configuration:

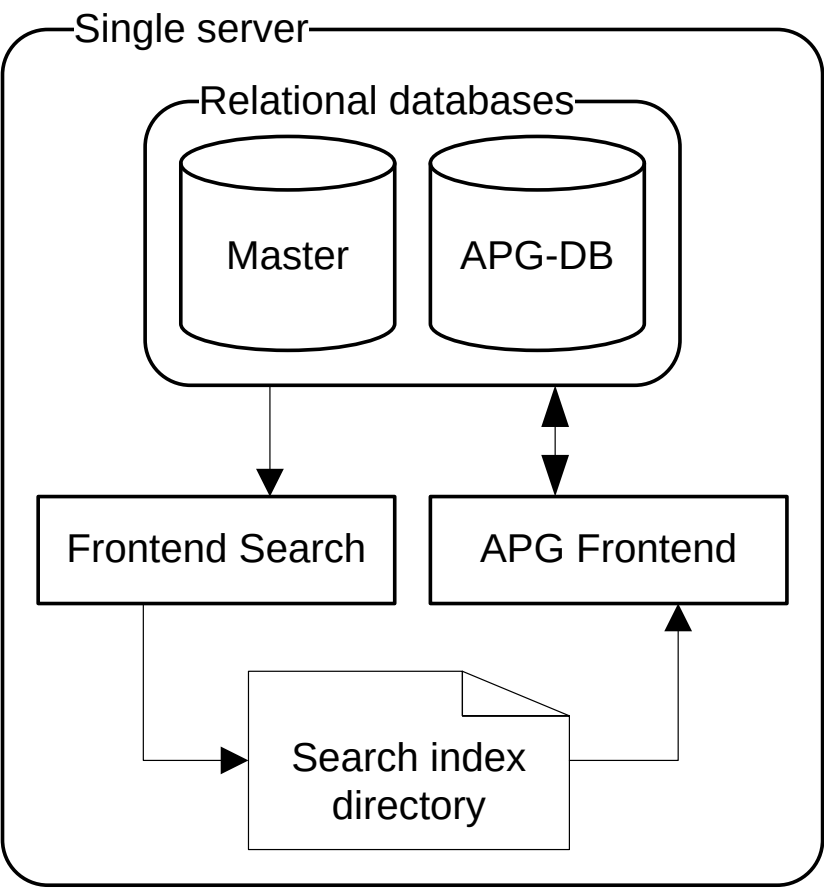


Figure 1: Single server configuration

Example of a distributed server configuration:

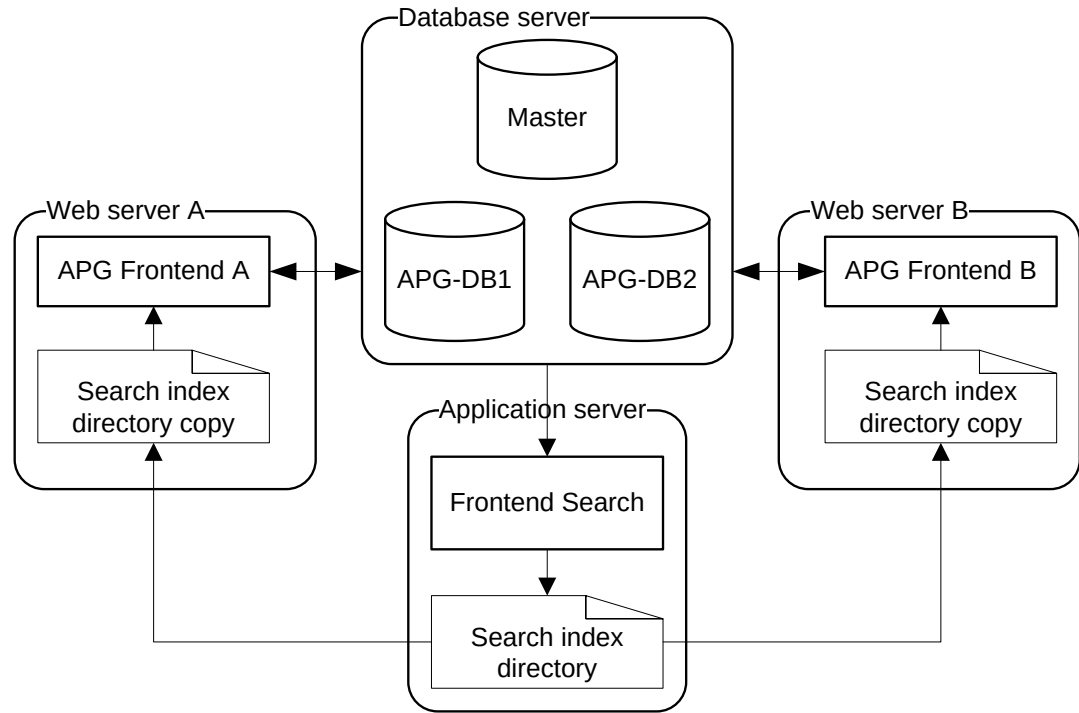


Figure 2: Distributed server configuration

3

Installation

Prerequisites The following operations *must* be performed before any installation or configuration steps are taken.

Check if the Frontend Search module is available in your APG installation:

- Use the Module Manager to *list* the *available* packages.
- Look for the *Filename* frontend-search-* and the *Module* Frontend-Search.
- If you can't find the Frontend Search module, you need to get the package frontend-search-*.pkg and place it in the APG Module Repository directory.
- Then, use the Module Manager again to check if the Frontend Search module is *available*.

Module installation Check if the Frontend Search module is installed:

- Use the Module Manager to *list* the *installed* packages.
- Look for the *Filename* frontend-search-* and the *Module* Frontend-Search.
- If you can't find the Frontend Search module, use the Module Manager to *install* the Frontend Search module.
- Then, use the Module Manager again to check if the Frontend Search module is *installed*.

4

Configuration

This section is about the configuration of the Frontend Search module. The configuration file is located in the `<APG Root>/Tools/Frontend-Search/<Instance>/conf` directory. In case this file does not exist, it must be created and should be stored in this folder as well.

Note: It is possible to change the location of the configuration file. It is useful if the frontend search is installed in another instance than the `Default` one. To change it, the java parameter `apg.frontend.search.config` must be add in `{unix|windows}-service.properties` file of the Tomcat configuration folder.

Example:

```
jvm.param.10=-Dapg.frontend.search.config=/APG/Tools/Frontend-Search/FS/conf/frontend-search.xml
```

Don't forget to update the tomcat service after adding that parameter.

You must use one Frontend Search instance for each APG Frontend instance. The configuration file is divided in two parts: one for the index generation process and one for the user searches handling.

`<configuration>`: The main container for the module configuration. Every other tag must be nested inside this one.

- `<index-directory>`: The directory where the index will be stored. You need to specify a full path.
Please note that the file storing the search index generation status (`*.state`) will have the same name as the index directory.
- `<index-configuration>`: the container for the index generation process settings. See section [Index configuration](#) page 10.
- `<search-configuration>`: the container for the user search handling settings. See section [Search configuration](#) page 13.

Please note that this is configuration file may be used by any APG modules providing the search capability like the APG Frontend.

1 Index configuration

- `<interval>` (optional): the duration (in seconds) between each index generation. 0 means that the index generation will only be run once. If set, the *interval* must be higher than the *duration*.
- `<duration>`: the duration (in seconds) allocated to the index generation. 0 means unlimited. If set, the *duration* must be lower than the *interval*.
- `<max-threads>`: the number of threads used for the search index generation.
- `<max-node-depth-logging>` (optional): the maximum node depth logging during the search index generation. Default value is 0 (no node will appear in the logs).
- `<working-directory>`: the directory used by the module to store temporary files. You can specify a full path or a relative path.
- `<indexed-values>`: the list of APG Frontend indexed values.
 - `<indexed-value key="key" fields="fields" analyzed="analyzed" force-from-key="keysrc" cached="cached" split-on-regex="regex">`: the indexed value properties.
 - * *key* (required): represent the APG Frontend report properties. See section [Indexed values key](#) page 12.

- * *fields* (required): represent the search index fields (separated by a ',') where the value should be stored. Most of the time, only one *field* will apply for one *key*. These fields should be in the <search-configuration> as <field> *name* attribute.
 - * *analyzed* (required): specifies if the value needs to be analyzed (cleaned and split) before adding it to the search index. For example, sentences are usually analyzed. IPs and device names are usually not analyzed.
 - * *keysrc* (optional): used to force the *key* property to be set even if not present. To find the *key* value, the search index generator will use *keysrc*.
 - * *cached* (optional): used for forced properties to know if the *key* values found have to be cached or need to be loaded all the time. It is recommended to enable cached properties to speed up the search index generation.
 - * *regex* (optional): used to split the property value before adding it to the search index. This *regex* value should be in the same as one of <indexed-field> *split-on-regex* attribute in the <search-configuration>. See section [Split indexed values and search keywords](#) page 14.
- <master>: the container for the master datasource used by the APG Frontend.
 - <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.
 - <apg>: the container for the APG datasources used by the APG Frontend.
 - <datasource id="*id*" driverClassName="*driverClassName*" url="*url*" username="*username*" password="*password*">: a performance datasource used by the APG Frontend
 - * *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.

1.1 Indexed values key

There are 2 kinds of indexed values key:

1. the node properties (starting with "node-").

List of available node properties key:

- *node-name*: the node name.
Example: "4. View per Device".
- *node-title*: the node title.
Example: "5. Topology Browser / 4. View per Device".
- *node-description*: the node description.
Example: "This report lists all devices with their properties (IP, Device Type, ...)".
- *node-breadcrumbs* : the node breadcrumbs.
Example: "All > Inventory > 5. Topology Browser > 4. View per Device".

2. the node property values (used in the node filter expression).

List of the most common node property values key:

- *device*: the device name.
- *ip*: the device IP.
- *name*: the variable name.
- *part*: the part name.
- *devtype*: the device type name.
- *parttype*: the part type name.

2 Search configuration

- `<indexed-fields>`: the list of fields present if the search index.
 - `<field name="name" analyzed="analyzed" boost="boost" split-on-regex="regex">`: the search index field properties.
 - * *name* (required): represents a search index field name. This field *name* should be in the `<index-configuration>` as one of `<indexed-value>` *fields* attribute.
 - * *analyzed* (required): specifies if the value need to be analyzed (cleaned and split) before searching it in the index. For example, sentences are usually analyzed. IPs and device names are usually not analyzed.
 - * *boost* (optional): represents the weight of the field in the search.
If the boost is less than 1.0, the field weight would be less important for the search.
If the boost is more than 1.0, the field weight would be more important for the search.
The default value is 1.0. The value must be greater than 0.0.
 - * *regex* (optional): used to split each search keyword in keywords before searching them in the index. The field *regex* should be in the same as one of `<indexed-value>` *split-on-regex* attribute in the `<index-configuration>`. See section [Split indexed values and search keywords](#) page 14.

3 Split indexed values and search keywords

To improve the results, some values need to be split before indexing them and/or searching them. To do that, you can use regex with the `split-on-regex` attribute in `<indexed-value>` (index) and `<field>` (search).

In the default configuration, the `device` value are split on "-" and "." with the regex "[\.-]";

- the "." is used to split FQDN device name.
Example: "server1.example.com" will be split in 3 values: "server1", "example" and "com".
- the "-" is used to split device name.
Example: "ca-montreal-mailserver" will be split in 3 values "ca", "montreal" and "mailserver".

5

Start Up

When the Frontend Search is installed and configured, the Frontend Search module is ready to start indexing the APG Frontend data.

To start the indexing, the `refresh-frontend-search` task must be run in the centralized management.

Another way to start the indexing is to run the script matching your operating system:

- in a Linux environment: `./bin/refresh-frontend-search.sh`
- in a Windows environment: `bin\refresh-frontend-search.cmd`

This command will launch the search index generation with the default configuration file:

`<APG Root>/Tools/Frontend-Search/<Instance>/conf/frontend-search.xml`.

If you want to use another configuration file, you need to use the `-c <conf-file>` option:

- in a Linux environment:

```
./bin/refresh-frontend-search.sh -c /my/directory/search-1.xml
```

- in a Windows environment:

```
bin\refresh-frontend-search.cmd -c "C:\my_directory\search-2.xml"
```

6

Filter nodes in the search index

It can be useful to skip nodes during the frontend search index generation to remove nodes that are not relevant to reduce the time required to complete the search index generation.

The node filtering is based on their *Unique Identifier* (UID) configurable in the APG Frontend (Edit Mode).

There are 2 strategies:

1. Include/Exclude specific nodes using one simple regex (like `*_NOINDEX`) and modifying the UIDs of nodes to match this regex.
2. Include/Exclude specific nodes using a complex regex (like `<NODE_UID1>|<NODE_UID2>`) to match existing node UIDs.

Note: Please note that you cannot explicitly include/exclude *Link nodes* since they do not have UIDs. Instead, for example, you can exclude a *link node*'s target, by filtering out all the children of the node it links to. Keep in mind, in this case, these child nodes will not be indexed if other nodes link to their parent. They will, on the other hand, be indexed if another link node links to them directly.

1 Include specific root nodes

If you want to make sure the index generation only includes specific root nodes, you can define the specific root nodes by using the `apg.search.index.include.tree` property configurable in the module configuration file:

```
<APG Root>/Tools/Frontend-Search/<Instance>/conf/module.properties
```

Root nodes with a UID matching the regex will be indexed.

Note: Please note that if the `apg.search.index.include.tree` property is used, root nodes without UIDs won't be indexed.

2 Exclude specific nodes

If you want to exclude any nodes from the index generation, you can exclude specific nodes by using the `apg.search.index.skip` property configurable in the module configuration file:

```
<APG Root>/Tools/Frontend-Search/<Instance>/conf/module.properties
```

All nodes with an UID matching the regex won't be indexed.

7

Schedule search index generation

There are three methods to schedule index generation:

- The first method uses a basic task scheduler included in the Frontend Search module. This method is not the preferred method, however, as it is not that configurable, and forces the service to run continually. Its advantage is that it is simple to set up.
- The second method uses the operating system's task scheduler (*cron* in Linux, the *Task Scheduler* in Windows).
- The third method uses the `refresh-frontend-search` task of the APG task scheduler. It is configurable in the centralized management. This method is the preferred one.

It is possible to set up more complex scheduling periods, which is useful to better accommodate for long indexing times.

Be aware that, especially for larger setups, the only way to ensure an always up-to-date search index with the latest reports is to perform the indexing continuously, possibly on a dedicated server (an example configuration is figure 2), or using another more complex configuration. Contact <https://>

[//support.emc.com](http://support.emc.com) for more information.

WARNING: It's recommended to run the `import-properties` task before refreshing the Frontend-Search index to improve performances.

1 Use the module scheduler

The Frontend Search module contains its own basic task scheduler to run the search index generation periodically.

Use the `<interval>` attribute in the module configuration file to set up the search index repetition.

It is recommended to use the `<duration>` attribute to set up a max duration lower than the `<interval>`.

Note that the `<interval>` tags do not appear in the configuration file by default.

See the [Index configuration](#) page 10.

2 Use the operating system task scheduler

1. In a Linux environment, you can use *cron*.

The Frontend Search script allows to schedule and unschedule the task every day at 4:00 AM:

```
./bin/refresh-frontend-search.sh [schedule|unschedule]
```

Please refer to the *cron* documentation for more details.

2. In a Windows environment, you can use the Windows *Task Scheduler*.

The Frontend Search script allows you to schedule and unschedule the task every day at 4:00 AM:

```
bin\refresh-frontend-search.cmd [schedule|unschedule]
```

Please refer to the *Task Scheduler* documentation matching your Windows version for more details.

3 Use the APG task scheduler

To be able to use the APG task scheduler, its service must be running. Check in the *Services* tab of the centralized management to manage the services.

By default, the `refresh-frontend-search` task is configured to run each day at 8:00 PM. It can be fully customized and multiple tasks can be created. See the *APG-Task-Scheduler.pdf* guide for more detail on how to create and modify a task.

4 Schedule multiple configurations for different days of the week

The indexing of the APG Frontend data can take hours and you may not want to run the full indexing every day, which occurs when using the default scheduled task configured during installation or in the above procedure.

You can split up the indexing to occur over the course of several days. The indexing is performed linearly through the report tree and is picked up where it left off if performed over several sessions. An impact of this is that, for example, if by mid-week the indexing has moved half way down the tree, and a report is added at the beginning of the tree, it will not be indexed until the indexing has cycled back to the beginning.

For typical environments, however, where there is not constant change in the reports available, simply distributing the indexing over the week, and performing a full indexing on the weekend, can yield good results, as explained in the example below.

Follow these steps to schedule a 1 hour indexing duration on every weekday with a full indexing every Sunday. (You can vary the duration, for example, to accommodate for larger databases.)

1. Create a new file `frontend-search-1h.xml` using the default configuration located in the `<APG Root>/Tools/Frontend-Search/<Instance>/conf` directory.
2. Edit the new file and set the `<duration>` to 3600 seconds (1 hour).
See section [Index configuration](#) page 10 for more details about the `<duration>` setting.
3. Schedule a new task.
 - Create a new task file `refresh-frontend-search-1h.task` using the default task file located in this directory:
`<APG Root>/Tools/Task-Scheduler/Default/data/task_repository/frontend-search/<Instance>/`

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

```

<task xmlns="http://www.watch4net.com/Tools/Scheduler/Tasks" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" name="refresh-frontend-search-1h-<Instance>" xsi:schemaLocation="http://www
.watch4net.com/Tools/Scheduler/Tasks task.xsd">

    <schedule cron="0 20 * * * " xsi:type="schedule-repeated" disabled="false"></schedule>

    <action xsi:type="java-action">
        <executing-class>com.watch4net.component.loader.util.ComponentLauncher</executing-class
    >
        <class-path>lib/apg-logging-jul.jar:lib/component-loader.jar</class-path>
        <execute-method>main</execute-method>
        <jvm-options>
            <option name="javaagent:">lib/apg-bootstrap-agent.jar</option>
            <option name="D">java.util.logging.config.file=conf/logging.properties</option>
            <option name="D">java.awt.headless=true</option>
            <option name="D">apg.database.accessor.provider=com.watch4net.apg.search.index.
SearchDedicatedDatabaseAccessorProvider</option>
        </jvm-options>
        <parameters>
            <param>index</param>
            <param>-c</param>
            <param><APG Root>/Tools/Frontend-Search/<Instance>/conf/frontend-search-1h.xml</
param>
        </parameters>
        <working-directory>.../Frontend-Search/<Instance>/.</working-directory>
    </action>

    <provider>
        <id>frontend-search</id>
        <instance-name><Instance></instance-name>
    </provider>

    <execution-locks>
        <lock>refresh-frontend-search-<Instance></lock>
    </execution-locks>

</task>

```

- Add a lock in the first task refresh-frontend-search.task to prevent the two tasks for from running at the same time, and change the cron of the task to only execute at 4:00 AM on sunday.

```
<?xml version="1.0" encoding="UTF-8"?>
<task xmlns="http://www.watch4net.com/Tools/Scheduler/Tasks" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" name="refresh-frontend-search-<Instance>" xsi:schemaLocation="http://www.
watch4net.com/Tools/Scheduler/Tasks task.xsd">

  <schedule cron="0 4 * * * 0" xsi:type="schedule-repeated" disabled="false"></schedule>
  ...
  <execution-locks>
    <lock>refresh-frontend-search-1h-<Instance></lock>
  </execution-locks>

</task>
```

- You should now see both tasks in the centralized management.

Advanced search

User search queries are processed by the Frontend Search module.
Here is the list of supported advanced search features:

- "+": all the search results will match the keyword.
Example: "abc xyz +router" will return results matching on "abc" and/or "xyz" and matching on "router".
- "-": exclude matching results from the unwanted keyword.
Example: "abc xyz -router" will return results matching on "abc" and/or "xyz" but not matching on "router".