

APS Default Discovery Service Provider

User Guide

Version: 1.0.0

Author: Tommy Svensson

Copyright © 2012 Natusoft AB

Table of Contents

1 APSDefaultDiscoveryServiceProvider	1
1.1 Discovery information	1
1.2 Transport data format	1
1.3 API	1

1 APSDefaultDiscoveryServiceProvider

This is a simple service where you can publish and unpublish service information. This information will be distributed to all other nodes that are configured to be part of the discovery.

Service data can be configured to be shared by any of the 3 TCP/IP protocols: Multicast, UDP, and TCP. The *APSTCPIPService* is used to send and receive data.

1.1 Discovery information

The discovery information is just a Properties object. Anything you want can be put into it, but the following keys are suggested for interoperability:

```
public class DiscoveryKeys {

    /** A name of the service. */
    public static final String NAME = "name";

    /** The version of the service. */
    public static final String VERSION = "version";

    /** An URI as used by APSTcpipService. */
    public static final String APS_URI = "apsURI";

    /** A URL for accessing the service. */
    public static final String URL = "url";

    /** The port of the service. */
    public static final String PORT = "port";

    /** The host of the service. */
    public static final String HOST = "host";

    /** An informative description of the service. */
    public static final String DESCRIPTION = "description";

    /** This is used by APSClusterService to announce cluster members. */
    public static final String APS_CLUSTER_NAME = "apsClusterName";

    /** The protocol of the service, like TCP, UDP, Multicast */
    public static final String PROTOCOL = "protocol";

    /** Some description of the type of the content provided by the service. */
    public static final String CONTENT_TYPE = "contentType";

    /** A timestamp of when the entry was last updated. */
    public static final String LAST_UPDATED = "lastUpdated";
}
```

1.2 Transport data format

The data is transported over the network in JSON format:

```
{
  action: "ADD" / "REMOVE",
  serviceDescription: {
    name: "myservice",
    version: "1.5.4",
    apsURI: "tcp://myhost:5564"
    ...
  }
}
```

1.3 API

```

/**
 * A network service discovery.
 *
 * There are many such services available in general, a bit less from a Java
 * perspective, but the intention with this is not to compete with any of
 * the others, but to provide an extremely simple way to discover remote
 * services in an as simple and primitive way as possible. Basically a way
 * to have multiple hosts running APS based code find each other easily,
 * may it be by simple configuration or by multicast or TCP, or wrapping
 * some other service.
 */
public interface APSSimpleDiscoveryService {

    //
    // Methods
    //

    /**
     * On a null filter all services are returned. The filter is otherwise
     * of LDAP type: (&(this=that)(something=pizza)).
     *
     * @param filter The filter to narrow the results.
     */
    Set<Properties> getServices(String filter);

    /**
     * Publishes a local service. This will announce it to other known
     * APSSimpleDiscoveryService instances.
     *
     * @param serviceProps This is a set of properties describing the
     *                      service. There are some suggested keys in
     *                      DiscoveryKeys for general compatibility.
     *
     * @throws APSDiscoveryException on problems to publish (note:
     *                               this is a runtime exception!).
     */
    void publishService(Properties serviceProps) throws APSDiscoveryException;

    /**
     * Recalls the locally published service, announcing to other known
     * APSSimpleDiscoveryService instances that this service is no longer available.
     *
     * @param unpublishFilter An LDAP type filter that matches an entry or entries
     *                        to unpublish. Any non locally published services caught
     *                        in the filter will be ignored.
     *
     * @throws APSDiscoveryException on problems to publish (note: this is a
     *                               runtime exception!).
     */
    void unpublishService(String unpublishFilter) throws APSDiscoveryException;
}

```