B

Oracle RAC Fast Application Notification

Starting from Oracle Database 12*c* Release 1 (12.1), the Oracle RAC Fast Application Notification (FAN) APIs provide an alternative for taking advantage of the high-availability (HA) features of Oracle Database, if you do not use Universal Connection Pool or Oracle WebLogic Server with Active Grid Link (AGL).

This appendix covers the following topics:

This feature depends on the Oracle Notification System (ONS) message transport mechanism. This feature requires configuring your system, servers, and clients to use ONS.

For using Oracle RAC Fast Application Notification, the `simplefan.jar` file must be present in the `CLASSPATH`, and either the `ons.jar` file must be present in the `CLASSPATH` or an Oracle Notification Services (ONS) client must be installed and running in the client system.

B.1 Overview of Oracle RAC Fast Application Notification

The Oracle RAC Fast Application Notification (FAN) feature provides a simplified API for accessing FAN events through a callback mechanism. This mechanism enables third-party drivers, connection pools, and containers to subscribe, receive and process FAN events. These APIs are referred to as Oracle RAC FAN APIs in this appendix.

The Oracle RAC FAN APIs provide FAN event notification for developing more responsive applications that can take full advantage of Oracle Database HA features. If you do not want to use Universal Connection Pool, but want to work with FAN events implementing your own connection pool, then you should use Oracle RAC Fast Application Notification.

> **Note:**
>
> - If you do not want to implement your own connection pool, then you should use Oracle Universal Connection Pool to get all the advantages of Oracle RAC Fast Application Notification, along with many additional benefits.
> - Starting from Oracle Database 12*c* Release 1 (12.1), implicit connection cache (ICC) is desupported. Oracle recommends to use Universal Connection Pool instead.

Your applications are enabled to respond to FAN events in the following way:

- Listening for Oracle RAC service down and node down events.

- Listening for Oracle RAC service up events representing any Oracle RAC or Global data Service (GDS) start or restart. For these UP events, FAN parameter status is `UP` and `event_type` is one of the following: `database`, `instance`, `service`, or `servicemember`.
- Supporting FAN ONS event syntax and fields for both Oracle Database Release 12*c* and earlier releases, for example, the `event_type` and `timezone` fields added to every event, the `db_domain` field added to every event other than node or public-network events, the `percentf` field added to Run-Time Load Balancing (RLB) events, and so on.
- Listening for load balancing advisory events and responding to them.

This feature exposes the FAN events, which are the notifications sent by a cluster running Oracle RAC, to inform the subscribers about an event happening at the service level or the node level. The supported FAN events are the following:

- Service up

  The service up event notifies the connection pool that a new instance is available for use, allowing sessions to be created on the new instance. The `ServiceUpEvent` Client API is supported in the current release of the Oracle RAC FAN APIs, that is, in the `simplefan.jar` file.

- Service down

  The service down events notify that the managed resources are down and currently not available for access. There are two types of service down events:

  – Events indicating that a particular instance of a service is down and the instance is no longer able to accept work.

  – Events indicating that all-but-one instances of a service are down and the service is no longer able to accept work.

- Node down

  The node down events notify that the Oracle RAC node identified by the host identifier is down and not reachable. The cluster sends node down events when a node is no longer able to accept work.

- Planned down

  Planned down events include all the down events, except node down event. These events have the following two fields set: `status=down` and `reason=user`.

- Load balancing advisory

  The load balancing advisory events provide metrics for load balancing algorithms. Load balancing advisories are sent regularly to inform subscribers of the recommended distribution of work among the available nodes.

> **Note:**
>
> If you want to implement your own connection pool, only then you should use Oracle RAC Fast Application Notification. Otherwise, you should use Oracle Universal Connection Pool to get all the advantages of Oracle RAC Fast Application Notification, along with many additional benefits.

Related Topics

- *Oracle Universal Connection Pool Developer's Guide*

B.2 Installing and Configuring Oracle RAC Fast Application Notification

You can install the Oracle RAC FAN APIs by performing the following steps:

1. Download the `simplefan.jar` file from the following link:

   https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html
2. Add the `simplefan.jar` file to the classpath.
3. Perform the following in your Java code:
   a. Get an instance of the `FanManager` class by using the `getInstance` method.
   b. Configure the event daemon using the `configure` method of the `FanManager` class. The `configure` method sets the following properties:

      `onsNodes`: A comma separated list of `host:port` pairs of ONS daemons that the ONS run time should communicate with. The `host` in a `host:port` pair is the host name of a system running the ONS daemon. The `port` is the local port configuration parameter for that daemon.

      `onsWalletFile`: The path name of the ONS wallet file. The wallet file is the path to a local wallet file used by TLS to store TLS certificates. Same as wallet file configuration parameter to ONS daemon.

      `onsWalletPassword`: The password for accessing the ONS wallet file.

> **See Also:**
>
> - For a detailed description of the Oracle RAC FAN APIs, refer to *Oracle Database RAC FAN Events Java API Reference*.
> - *Oracle Universal Connection Pool Developer's Guide*

B.3 Using Oracle RAC Fast Application Notification

The following code snippet explains how to handle FAN down events. This example code prints the event data to the standard output device.

This example code demonstrates how to use Oracle RAC FAN APIs by overloading the `handleFanEvent` method to accept different FAN event notifications as arguments. The example code also displays event data such as:

- Name of the system sending the FAN event notification
- Timestamp of the FAN event notification
- Load status of the FAN event notification

Example B-1 Example of Sample Code Using Oracle RAC FAN API for FAN Down Events

```
...
...Properties props = new Properties();
```

```
props.putProperty("serviceName", "gl");
FanSubscription sub = FanManager.getInstance().subscribe(props);
sub.addListener(new FanEventListener()) {
  public void handleFanEvent(ServiceDownEvent se) {
    try {
      System.out.println(event.getTimestamp());
      System.out.println(event.getServiceName());
      System.out.println(event.getDatabaseUniqueName());
      System.out.println(event.getReason());
      ServiceMemberEvent me = se.getServiceMemberEvent();
      if (me != null) {
        System.out.println(me.getInstanceName());
        System.out.println(me.getNodeName());
        System.out.println(me.getServiceMemberStatus());
      }
      ServiceCompositeEvent ce = se.getServiceCompositeEvent();
      if (ce != null) {
        System.out.println(ce.getServiceCompositeStatus());
      }
    }
    catch (Throwable t) {
      // handle all exceptions and errors
      t.printStackTrace(System.err);
    }
  }
  public void handleFanEvent(NodeDownEvent ne) {
    try {
      System.out.println(event.getTimestamp());
      System.out.println(ne.getNodeName());
      System.out.println(ne.getIncarnation());
    }
    catch (Throwable t) {
      // handle all exceptions and errors
      t.printStackTrace(System.err);
    }
  }
  public void handleFanEvent(LoadAdvisoryEvent le) {
    try {
      System.out.println(event.getTimestamp());
      System.out.println(le.getServiceName());
      System.out.println(le.getDatabaseUniqueName());
      System.out.println(le.getInstanceName());
      System.out.println(le.getPercent());
      System.out.println(le.getServiceQuality());
      System.out.println(le.getLoadStatus());
    }
    catch (Throwable t) {
      // handle all exceptions and errors
      t.printStackTrace(System.err);
    }
  }
});
```

Example B-2 Example of Sample Code Using Oracle RAC FAN API for FAN Up Events

The following code snippet explains how to use Oracle RAC Fast Application Notification for service up events. The code uses the new Oracle RAC FAN APIs introduced in Oracle Database 12*c* Release 2 (12.2.0.1), namely, `FanUpEventListener` interface that extends the `FanEventListener` interface and `ServiceUpEvent`. You must implement the

`FanUpEventListener` interface in the client application in the similar way you implement the `FanEventListener` interface.

```
import oracle.simplefan.*;

...
FanEventListener fanListener = new FanUpEventListener() {
 public void handleEvent(ServiceUpEvent event) { ..... }

// Specify the next action here, when the node comes up
public void handleEvent(NodeUpEvent event) { ..... }
    ......
}
FanManager fanMngr = FanManager.getInstance();
Properties onsProps = new Properties();
onsProps.setProperty("onsNodes", ......);
fanMngr.configure(onsProps);

Properties subscriptionProps = new Properties();
subscriptionProps.setProperty("serviceName", ......);
fanSubscription = fanMngr.subscribe(subscriptionProps);
fanSubscription.addListener(fanListener);
...
```

B.4 Implementing a Connection Pool

You must implement your own connection pool for using Oracle RAC FAN APIs. Consider the following points before you implement a connection pool using the Oracle RAC FAN APIs:

- Oracle RAC FAN APIs provide a subset of FAN events.
- Oracle RAC FAN APIs support only ONS events. If you want your application to support corresponding supercluster events, then you may require additions to the subscription properties.