

Oracle Technologies Primer

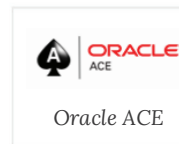
An Oracle Fusion Middleware and iPaaS blog!



[Home](#) [About Your Author](#) [FMW/PaaS Community Contribution](#)

Tag Archives: Retrying Faults in SOA Suite

Oracle ACE



Email Subscription

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Join 1,185 other followers

Fault Handling in Oracle SOA Suite : Advanced Concepts

Posted on [July 18, 2011](#) by [Arun Pareek](#)

This tutorial is meant to cover extensively the mechanism that we can adopt for Fault Management for a SOA Suite composite. It will deal with a fairly overall strategy for handling faults and dealing with them in various ways.

Before diving more into advanced concepts of Handling Faults let me present a small introduction covering the basics of a Service composite.

Basic Architecture of a Service Composite in Oracle SOA Suite

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use. To find out more, including how to control cookies, see here: [Cookie Policy](#)

Port Type in
Oracle PCS
Accelerating
Innovation
through Device,
Human and
Process
Centric Integrati
on
Suppress
Approval
Controls in
Oracle BPM –
Hidden
Workspace Featu
re

Blog Stats

551,172 hits

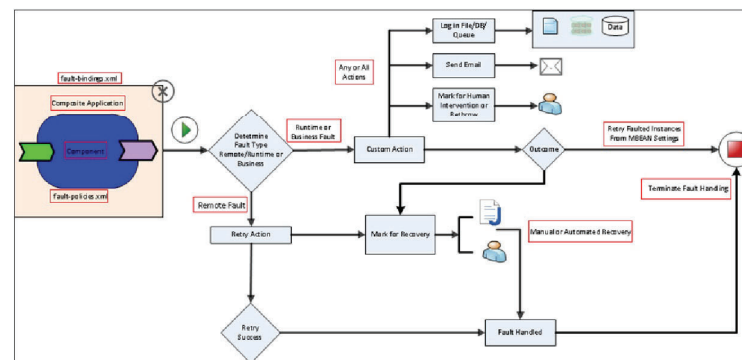
Top Posts

Change Logging
Level of
Integrated Weblo
gicServer in
JDeveloper
Stripping XML
Namespaces
using XQuery
Dynamic XQuery
in Oracle Service
Bus
Oracle BPM
Human Task
Management :
Using Human
Task Events to
Invoke Microsoft
Exchange Web
Services
Eliminating

1. The **fault condition** that activates the policy block—we specify what type of fault(s) the policy is relevant for. We can then apply even more finer grained policy and actions based on error codes, error messages etc.
2. The **action(s)** that should be performed when the condition is satisfied. An action for an fault may be to retry it for a certain number of time at a specified interval, or to mark it in recovery for human intervention, use a custom Java code or simply to throw the fault back. If the fault is rethrown then if we have specified any explicit '**catch**' block in our BPEL process that will be executed.

It should also be noted that fault policies need to be explicitly associated with composites, components, or references. This is done in a **fault-bindings.xml** file. Fault bindings link the composite, specific components in the composite, or specific references in the components on the one hand to one of the fault policies on the other.

Have a look at the diagram below to understand a mechanism to throw a fault from a service composite, identify the fault type and then take necessary action.



The following post will try and cover all aspects of what is covered in the diagram above.

Consider the following fault-policies.xml. Read the comments in the XML to understand what each of the condition, action and policy is about.

```

1 <faultPolicies xmlns="http://schemas.oracle.com/b
2 <faultPolicy version="2.0.1" id="CompositeFaultPo
3 <Conditions>
4 <!-- Conditions can be fine grained to include Ac
5 <faultName xmlns:bpelx="http://schemas.oracle.com

```