



SelfDiagnose

“who is to blame”

ernest micklei, April 2007



Overview

- Motivation
- Introduction
- Demo
- Quick steps
- Expert use



Motivation

- Some application problems are related to:
 - Configuration
 - Availability

- Logging not always pinpoints the cause
 - level (debug)
 - exception encapsulation
 - “only when it happens”



Problem causes

- Non-matching names in JNDI
- Absent properties, files
- Missing bindings in JNDI
- No access to database, queue, host
- Missing classes on runtime classpath
- Missing System properties
-



Problem revealings

- ❑ An error has occurred
- ❑ SEVERE Could not start /myapp
- ❑ DB2 SQL-ERROR 10493
- ❑ Log4j shutting down



Who wants to know?

- Development
 - Is my configuration correct?
- Application Server admin
 - Is their configuration correct?
 - Is my deployment script correct?
- Maintenance
 - Are all (detectable) resources available?
- Functional Maintenance



SelfDiagnose

- ❑ SelfDiagnose is a library of diagnostic tasks that can verify the availability of resources

- ❑ Application “itself” runs diagnose

- ❑ Open-source Java project since 2006
 - selfdiagnose.sourceforge.net



Task

- A DiagnosticTask verifies the availability of a resource required for the execution of an application

- Availability
 - Deployment (configuration only)
 - Runtime (execution only)



Example: Property

- Configuration
 - key present?
 - does value matches pattern?
- Availability
 - classpath, JNDI

```
<checkresourceproperty  
  name="environment.properties"  
  property="ProviderStoreDB4O.bin" />
```



Example: Datasource

- ❑ Configuration
 - JNDI name, Driver, User, Access
- ❑ Available
 - host reachable?, connectable?

```
<checkdatasourceconnectable  
  name="jdbc/oracle/ubs_userBeheerService-ubsDs"/>
```



Example: Table exists

- ❑ Configuration
 - Table name
- ❑ implies CheckDataSourceConnectable

```
<checkdatabasetableexists  
  name="UBS_CONTEXT"  
  datasource="jdbc/oracle/ubs_userBeheerService-ubsDs"  
  database="oracle" />
```



Example: JMS

JMS Queue

■ Configuration

- JNDI name, Queue name, CCSID

■ Availability

- connectable?

```
<checkjndibinding  
  name="jms/mq/ubs_userBeheerServiceQCf"  
  class="javax.jms.QueueConnectionFactory"/>
```



Example: URL

- Host reachable?
 - Glue servlet
 - mailserver
 - other Webservice
 - ...

```
<checkurlreachable url="http://s3browse.com"/>
```



Example: JMX

- ❑ Managed Bean property
 - value set?
 - does value matches pattern?
- ❑ Dangerous
 - (if accessible at all)

```
<checkmbeanproperty
  mbeanfactory="com.....AdminServiceFactory"
  query="WebSphere....ontvangenBLIBerichtLp,*"
  method="isStarted"
  pattern="true" />
```



Example: MDB

- EJB Message Driven Bean
 - Configuration
 - JNDI
 - Availability
 - process running?
- (currently IBM WebSphere 5+ only)

```
<checkwebspherejmslistenerport name="ubs_userBeheerServiceLp"/>
```



Design

- DiagnosticTask
- Configuration
- Reporting



DiagnosticTask

- Initialize
 - Load configuration parameters
- Setup
 - Verifiy parameters
- Run
 - Execute check using parameters
 - Report to DiagnosticTaskResult



Configuration

□ XML

```
<selfdiagnose
  xmlns="http://selfdiagnose.sourceforge.net"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://selfdiagnose.sourceforge.net
  http://selfdiagnose.sourceforge.net/selfdiagnose.xsd" >
  <tasks>
    <!-- add your tasks here -->
  </tasks>
</selfdiagnose>
```

□ Java

```
private final static String MAPS_URL =
    Check.property("environment.properties", "google.maps.url");
```



Reporting

□ Log4j

```
2007-Apr-16 11:03:09.050 [...] INFO SelfDiagnose - JNDI binding [url/config/ubs_userBeheerService/config] of type [java.net.URL] is available in naming server - /appl/was00t/WebSphere/AppServer/installedApps/was00tcell/ubs_userBeheerService.ear/ubs_UserBeheerServiceWeb.war/selfdiagnose.xml
```

□ SelfDiagnoseServlet

- remote invocation of `SelfDiagnose.run()`
- Formats: HTML, XML, plain



Reporting II

- for each result
 - Task name
 - Status {passed,failed,error}
 - Message
 - Requestor
 - = what component added the task



Quickstart

- Add Maven dependency
- Add Servlet to Web Descriptor
- Define selfdiagnose.xml



Maven dependency

□ project.xml

```
<dependency>
  <groupId>com.philemonworks</groupId>
  <artifactId>selfdiagnose</artifactId>
  <version>1.0</version>
  <type>jar</type>
  <properties>
    <war.manifest.classpath>>true</war.manifest.classpath>
    <war.bundle>>true</war.bundle>
  </properties>
</dependency>
```



Servlet Descriptor

```
<!-- ===== SELF DIAGNOSE CONFIGURATION ===== -->
<servlet>
  <servlet-name>selfdiagnose</servlet-name>
  <display-name>selfdiagnose</display-name>
  <servlet-class>
    com.philemonworks.selfdiagnose.SelfDiagnoseServlet
  </servlet-class>
</servlet>

<!-- ===== SELF DIAGNOSE CONFIGURATION ===== -->
<servlet-mapping>
  <servlet-name>selfdiagnose</servlet-name>
  <url-pattern>/selfdiagnose</url-pattern>
</servlet-mapping>
```



selfdiagnose.xml (example)

```
<selfdiagnose>
  <tasks>
    <checkdatasourceconnectable
      name="jdbc/oracle/ubs_userBeheerService-ubsDs"/>
    <checkdatabasetableexists
      name="UBS_CONTEXT"
      datasource="jdbc/oracle/ubs_userBeheerService-ubsDs"/>
    <checkjndibinding
      name="jms/mq/ubs_userBeheerServiceQCf"
      class="javax.jms.QueueConnectionFactory"/>
    <checkwebspherejmslistenerport name="ubs_userBeheerServiceLp"/>
    <checkjndibinding name="mail/MailService"/>
  </tasks>
</selfdiagnose>
```




Expert use

- Variables
- Custom Reports
- Custom task



Variables

- ❑ Capture result of a task
- ❑ Do other checks on that result

```
<checkjndibinding  
  name="url/config/ubs_userBeheerService/config"  
  class="java.net.URL"  
  var="config"/>
```

```
<checkurlreachable url="${config}"/>
```



Custom Reporting

- XML report + custom XSL

```
/selfdiagnose?format=xml&xsl=xsl/selfdiagnose.xsl
```



Custom Task

```
package com.mybusiness.myproduct;
public class MyTask extends DiagnosticTask {

    public String getDescription() { ... }

    public void initialize(Attributes attributes) { ... }

    public void run(ExecutionContext ctx, DiagnosticTaskResult result) { ... }
}
```

```
<task
  class="com.mybusiness.myproduct.MyTask">
  my-parameter1="my-value1"
/>
```



Example: CheckWSDL

```
public class CheckWSDL extends DiagnosticTask {
    private String url;

    public String getDescription(){
        return "checks that an URL refers to a WSDL";
    }
    public void initialize(Attributes attributes) {
        url = attributes.getValue("url");
    }
    public void run(ExecutionContext ctx, DiagnosticTaskResult result) {
        URL newURL = new URL(url);
        URLConnection newCon = newURL.openConnection();
        ...
        result.setPassedMessage(
            "URL ["+newURL+"] is reachable and " +
            "WSDL content could be retrieved.");
    }
}
```



CheckWSDL config

```
<task  
  class="CheckWSDL">  
  url="/context/Money4NothingService.wsdl"  
</task>
```



Testing SelfDiagnose

- Is my configuration correct?
- Add SelfTest class to your TestSuite



Wrapup

- SelfDiagnose provides simple means for:
 - Auto check of configuration
 - Auto check availability of resources
 - On demand report on “condition” of application



Contribute

- Improve tasks + new
- Documentation
- More tests
- Dashboard

<http://selfdiagnose.sourceforge.net/>

mailto: ernest.micklei@philemonworks.com



UML

