

## Deploying to Multiple Environments with Different Configuration Values

When the testing and development phase of a BPEL process has completed, you are ready to deploy the process to a production environment. The configuration properties for the production environment are typically different from those for the development environment. For example, the partner link Web service URL for the production environment is typically different from the one used in the testing environment.

In previous releases, you resolved this conflict by modifying the `bpel.xml` deployment descriptor file or providing separate `bpel.xml` files for the development and production environments. This was a big effort if the `bpel.xml` file required many modifications or if you had multiple versions of `bpel.xml` that required modifications.

Beginning with this release, a `customize` ant task is provided that enables you to specify the property value for development and production environments in a single build file location. The `customize` task captures the changes between different versions of `bpel.xml` using the build file. This task can be used as a subtask of `bpelc` or as a standalone ant task. The `bpelc` subtask can contain multiple `customize` tasks. The `customize` task can contain multiple subtasks.

### customize ant Task Example

This section provides an end-to-end example of using the `customize` ant task.

1. Copy the target domain URL WSDL file of the partner link Web service to the directory in which the `bpel.xml` file for your BPEL process is located (for this example, `IncrementService.wsdl` is used).
2. Open the `bpel.xml` file in the same directory.
3. Modify the directory path for the `IncrementService.wsdl` file under the `partnerLinkBinding` element to indicate that this WSDL file is located in the same directory:

```
<?xml version="1.0" encoding="UTF-8"?>
<BPELSuitcase>
  <BPELProcess src="Invoke.bpel" id="Invoke">
    <partnerLinkBindings>
      <partnerLinkBinding name="client">
        <property name="wsdlLocation">Invoke.wsdl</property>
      </partnerLinkBinding>
      <partnerLinkBinding name="IncrementService">
        <property name="wsdlLocation">IncrementService.wsdl</property>
      </partnerLinkBinding>
    </partnerLinkBindings>
    <configurations>
      <property name="testIntroduction"><![CDATA[
This BPEL process invokes a synchronous integer increment service.
]]></property>
    </configurations>
  </BPELProcess>
</BPELSuitcase>
```

4. Create a custom `build.xml` file in the *same* directory as the `bpel.xml` file and the `IncrementService.wsdl` file that uses the `customize` task. You specify properties for the host name, port, domain name, and process revision of the partner link Web service that map to values set in the `ant-orapbel-properties` file.

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```

<project name="bpel.deploy" default="compile" basedir=".">
  <target name="compile">
    <bpelc input="${basedir}/bpel.xml" out="${process.dir}/output"
      rev="${rev}" home="${bpel.home}">
      <customize>
        <partnerLinkBinding name="IncrementService">
          <property name="wsdlLocation">
            http://${host_name}:${port_number}
              /orabpel/${domain_name}
              /IncrementService/${rev}/IncrementService?wsdl</property>
          </partnerLinkBinding>
        </customize>
      </bpelc>
    </target>
  </project>

```

---

**Note:** You can use your own custom build properties file instead of the ant-orabpel-properties file. Ensure that you import the custom file in the build.xml file.

---

- Specify values for the host name, port, domain name, and process revision properties in the *SOA\_Oracle\_Home\bpel\utilities\ant-orabpel-properties* file.

#Development or production environment

```

host_name=jsmith-pc.us.oracle.com
port_number=80
domain_name=default
rev=1.0

```

```

# -----
# END OF FILE

```

This single file is the only location that requires editing between the development and production environments. For example, if the host name for the production environment is different (for example, jwilliams-pc.us.oracle.com), you change it in this file.

- Go into the main build.xml file located one directory *above* the bpel.xml, IncrementService.wsdl, and custom build.xml files.
- Specify a new target name in the process-deploy section (for this example, the default target name of compile is changed to compile1).

```

<target name="process-deploy"
  depends="validateTask, compile1, deployProcess, deployTaskForm,
  deployDecisionServices" />

```

- Create a new target name section below the target name. For this example, a section with a value of compile1 is specified that calls the custom build.xml script you created in Step 4. The target name section points to the bpel directory that includes the bpel.xml, IncrementService.wsdl, and custom build.xml files.

```

<target name="process-deploy"
  depends="validateTask, compile1, deployProcess, deployTaskForm,
  deployDecisionServices" />
  <target name="validateTask">

```

```

        <echo>
-----
| Validating workflow
-----
        </echo>
        <validateTask dir="${process.dir}/bpel" />
    </target>
    <target name="compile1">
        <ant dir="${process.dir}/bpel"/>
    </target>
    <target name="compile">

```

9. Deploy the BPEL process with ant.

```
ant process-deploy
```

10. Go to Oracle BPEL Control to see that all processes deployed correctly.

### customize ant Task Syntax

The `customize ant` task can be used to modify several types of properties. This section provides syntax examples of using the `customize ant` task with these properties.

**Specifying Custom File Names** By default, the `bpel.xml` file set in the `build.xml` file is used. You can also specify custom input and output deployment descriptor files. This action modifies the specified input XML file and writes the results to the specified `outFile`. Table 19–1 describes the attributes to use.

**Table 19–1 Attributes**

Attribute	Description	Required
<code>inFile</code>	Specifies the input XML file. When used as a subtask of <code>bpelc</code> , the default value is the deployment descriptor file ( <code>bpel.xml</code> ).	No
<code>outFile</code>	Specifies the output XML file. When used as a subtask of <code>bpelc</code> , the default value of this is the deployment descriptor file ( <code>bpel.xml</code> ).	No

**Specifying Deployment Descriptor configurations Properties** When used as a `bpelc` subtask, the `customize ant` task can add or modify configurations properties of the deployment descriptor.

```

<configurations>
    <property name="propName">propValue</property>
</partnerLinkBinding>

```

**See Also:** Appendix C, "Deployment Descriptor Properties" of *Oracle BPEL Process Manager Developer's Guide*

**Specifying partnerLinkBinding Properties** When used as a `bpelc` subtask, the `customize ant` task can add or modify properties of a `partnerLinkBinding`.

```

<partnerLinkBinding name="partnerLinkName">
    <property name="propName">propValue</property>
</partnerLinkBinding>

```

**See Also:** "customize ant Task Example" on page 19-6

**Specifying activationAgent Properties** When used as a `bpelc` subtask, the `customize ant` task can add or modify properties of an `activationAgent`.

```
<activationAgents name="actAgentName">
  <property name="propName">propValue</property>
</partnerLinkBinding>
```

**Specifying Deployment Descriptor preferences Properties** When used as a `bpelc` subtask, the `customize ant` task can add or modify preferences properties of the deployment descriptor.

```
<preferences>
  <property name="propName">propValue</property>
</preference>
```

**See Also:** Appendix C, "Deployment Descriptor Properties" of *Oracle BPEL Process Manager Developer's Guide*