

XService Workflow Engine 用户手册

本手册通过一个示例流程来介绍流程的开发、部署及调用

示例流程

该示例流程名为“example”，该流程接收一个字符串的输入，然后去调用一个名为“toLowerCase”的 Web 服务将字符串中的大写字母转换成小写字母，最后流程将 Web 服务调用的结果返回给客户端。流程如图 1 所示：

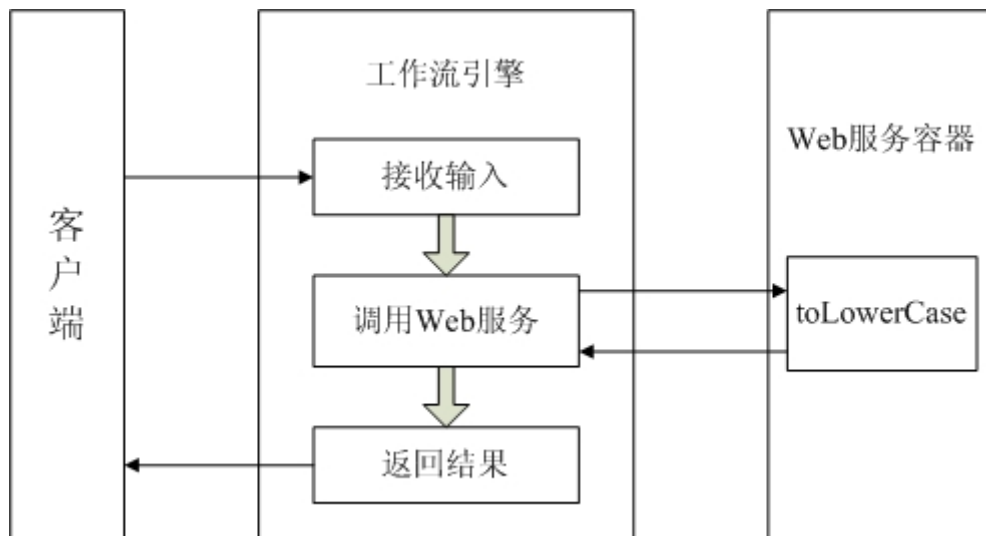


图 1 示例流程

Web 服务

在示例中调用的 Web 服务“toLowerCase”将输入字符串的大写字母变成小写字母，然后输出，其内部逻辑为：

```
public class toLowerCase {
    public String getLowerCase(String str)
    {
        return str.toLowerCase();
    }
}
```

toLowerCase 服务部署在 Web 服务容器中，如 Axis、WebLogic、ACT XService Container 等，在 %Engine_HOME%\example\webservice 目录下，我们提供了该服务在 ACT XService Container 中的部署单元“toLowerCase.wsar”。

在流程设计时，需要 toLowerCase 服务的 WSDL，服务的 WSDL 可以通过 Web 服务容器自带的 JAVA2WSDL 工具生成，也可以手写，toLowerCase.wsdl 如下：

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<wsdl:definitions targetNamespace="http://act.org" xmlns:tns=http://act.org
    xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    <wsdl:message name="getLowerCaseRequest">
        <wsdl:part name="param0" type="xsd:string"/>
    </wsdl:message>
    <wsdl:message name="getLowerCaseResponse">
        <wsdl:part name="getLowerCaseReturn" type="xsd:string"/>
    </wsdl:message>
    <wsdl:portType name="toLowerCase">
        <wsdl:operation name="getLowerCase" parameterOrder="param0">
            <wsdl:input name="getLowerCaseRequest" message="tns:getLowerCaseRequest"/>
            <wsdl:output name="getLowerCaseResponse" message="tns:getLowerCaseResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="toLowerCaseSOAPBinding" type="tns:toLowerCase">
        <wsdlsoap:binding style="rpc"
            transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="getLowerCase">
            <wsdlsoap:operation soapAction=""/>
            <wsdl:input name="getLowerCaseRequest">
                <wsdlsoap:body use="encoded"
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                    namespace="http://act.org"/>
            </wsdl:input>
            <wsdl:output name="getLowerCaseResponse">
                <wsdlsoap:body use="encoded"
                    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                    namespace="http://act.org"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="toLowerCaseService">
        <wsdl:port name="toLowerCase" binding="tns:toLowerCaseSOAPBinding">
            <wsdlsoap:address location=" http://localhost:8090/toLowerCaseService" />
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>

```

其中 <http://localhost:8090/toLowerCaseService> 为 Web 服务部署的地址，可根据实际情况进行修改。

流程描述

角色定义

我们首先在 example_ext.wsdl 中定义 example 流程与 Web 服务 toLowerCase 之间的角色关系，如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace=http://your.target.namespace/example
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ns1="http://www.act.org/userdefined" xmlns:ns2="http://act.org"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link"/>
  <import location="example.wsdl" namespace="http://www.act.org/userdefined"/>
  <import location="toLowerCase.wsdl" namespace="http://act.org"/>
  <plnk:partnerLinkType name="recToWorkflow">
    <plnk:role name="recRole">
      <plnk:portType name="ns1:rec"/>
    </plnk:role>
    <plnk:role name="recRole">
      <plnk:portType name="ns1:rec"/>
    </plnk:role>
  </plnk:partnerLinkType>
  <plnk:partnerLinkType name="lowerCaseToWorkflow">
    <plnk:role name="toLowerCaseRole">
      <plnk:portType name="ns2:toLowerCase"/>
    </plnk:role>
    <plnk:role name="recRole">
      <plnk:portType name="ns1:rec"/>
    </plnk:role>
  </plnk:partnerLinkType>
</definitions>
```

接口定义

然后，我们在 example.wsdl 中定义流程对外的接口，如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="http://www.act.org/userdefined"
  xmlns=http://schemas.xmlsoap.org/wsdl/
  xmlns:toi=http://www.act.org/userdefined
  xmlns:wsdlsoap=http://schemas.xmlsoap.org/wsdl/soap/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:bpws = "http://schemas.xmlsoap.org/ws/2003/03/business-process/">
  <types>
    <schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"/>
  </types>
```

```

<message name="recRequest">
  <part name="recInput" type="xsd:string"/>
</message>
<message name="recResponse">
  <part name="recOutput" type="xsd:string"/>
</message>
<portType name="rec">
  <operation name="rec">
    <input message="toi:recRequest" name="recInput"/>
    <output message="toi:recResponse" name="recOutput"/>
  </operation>
</portType>
<binding name="rec" type="toi:rec">
  <wsdlsoap:binding required="true" style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="rec">
    <wsdlsoap:operation soapAction=""/>
    <input>
      <wsdlsoap:body encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
        use="encoded"/>
    </input>
    <output>
      <wsdlsoap:body encodingStyle=http://schemas.xmlsoap.org/soap/encoding/
        use="encoded"/>
    </output>
  </operation>
</binding>
</definitions>

```

属性定义

接下来，我们在 example_property.wsdl 中定义流程中需要使用的一些属性，如下所示：

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace=http://org.act.properties
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:ns1="http://www.act.org/userdefined" xmlns:ns2="http://act.org">
  <bpws:property name="lowCaseProp" type="xsd:string"/>
  <bpws:propertyAlias messageType="ns2:getLowerCaseRequest" part="param0"
    propertyName="tns:lowerCaseProp"/>
</definitions>

```

流程定义

最后，我们在 example.bpel 中定义流程的各个环节，如下所示：

```
<?xml version="1.0" encoding="gb2312"?>
<process name="example" targetNamespace="http://your.target.namespace/example"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  queryLanguage="http://www.w3.org/TR/1999/REC-xpath-19991116"
  expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-19991116"
  suppressJoinFailure="no"
  enableInstanceCompensation="no"
  abstractProcess="yes"
  xmlns:tns="http://your.target.namespace/process"
  xmlns:props="http://org.act/properties"
  xmlns:ns1="http://act.org"
  xmlns:ns2="http://www.act.org/userdefined">
  <partnerLinks>
    <partnerLink name="recPL" partnerLinkType="tns:recToWorkflow" myRole="recRole"
      partnerRole="recRole"/>
    <partnerLink name="toLowerCasePL" partnerLinkType="tns:lowerCaseToWorkflow"
      myRole="recRole" partnerRole="toLowerCaseRole"/>
  </partnerLinks>
  <variables>
    <variable name="recIn" messageType="ns2:recRequest"/>
    <variable name="recOut" messageType="ns2:recResponse"/>
    <variable name="lowerCaseIn" messageType="ns1:getLowerCaseRequest"/>
    <variable name="lowerCaseOut" messageType="ns1:getLowerCaseResponse"/>
  </variables>
  <correlationSets>
    <correlationSet name="id" properties="props:lowerCaseProp"/>
  </correlationSets>
  <sequence name="" joinCondition="OR" suppressJoinFailure="no">
    <receive name="receive1" joinCondition="OR" suppressJoinFailure="no"
      partnerLink="recPL" portType="ns2:rec" operation="rec" variable="recIn"
      createInstance="yes">
    </receive>
    <assign name="assign1" joinCondition="OR" suppressJoinFailure="no">
      <copy>
        <from variable="recIn" part="recInput"/>
        <to variable="lowerCaseIn" part="param0"/>
      </copy>
    </assign>
    <invoke name="invoke toLowerCase" joinCondition="OR" suppressJoinFailure="no"
      partnerLink="toLowerCasePL" portType="ns1:toLowerCase"
      operation="getLowerCase" inputVariable="lowerCaseIn"
      outputVariable="lowerCaseOut">
    </invoke>
  </sequence>
  <correlations>
```

```

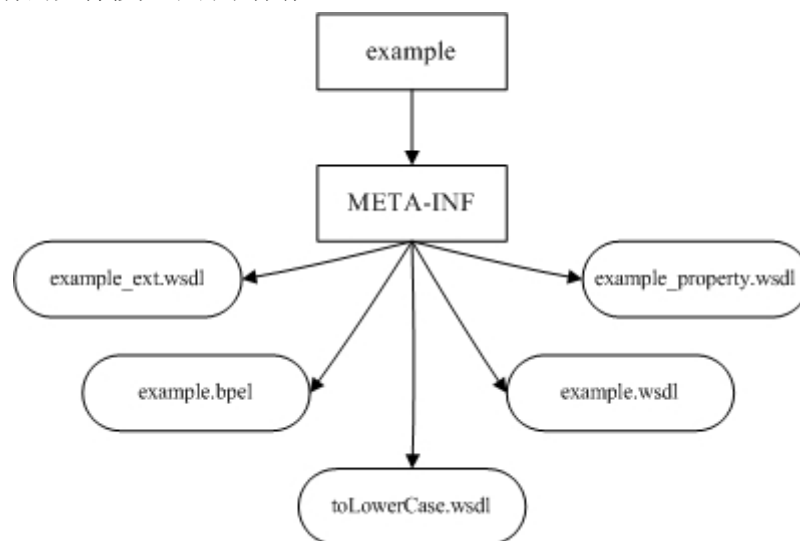
    <correlation set="id" initiate="yes" pattern="out-in"/>
  </correlations>
</invoke>
<assign name="assign2" joinCondition="OR" suppressJoinFailure="no">
  <copy>
    <from variable="lowerCaseOut" part="getLowerCaseReturn"/>
    <to variable="recOut" part="recOutput"/>
  </copy>
</assign>
<reply name="" joinCondition="OR" suppressJoinFailure="no"
  partnerLink="recPL" portType="ns2:rec" operation="rec" variable="recOut"/>
</sequence>
</process>

```

流程打包

流程描述完成后，我们需要将所有的文件打包成 XService Workflow Engine 的部署单元（.bar），下面是打包过程（在%Engine_HOME%\example\process 目录下，我们提供了打包完成的 example.bar）。

首先，将所有的文件按如下目录保存：



然后，在 windows 下执行下列命令打包生成 example.bar

```

C:\> cd example
C:\ example> jar cf example.bar META-INF

```

流程部署

通过 XService Workflow Manager 或者直接将 example.bar 复制到%Engine_HOME%\deploy 目录中，重启 XService Workflow Engine 即可。

客户端

客户端开发

我们使用 Axis 提供的 Web 服务客户端类包开发调用 example 流程的客户端 ExampleClient.java（位于%Engine_HOME%\example\client中），代码如下：

```
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import javax.xml.namespace.QName;

public class ExampleClient {
    public static void main(String[] args) {
        try {
            String endpoint = "http://localhost:8091/example";
            Service service = new Service();
            Call call = (Call) service.createCall();
            call.setTargetEndpointAddress( new java.net.URL(endpoint) );
            call.setOperationName(new QName("", "rec"));
            String ret = (String) call.invoke( new Object[] { "ABCD" } );
            System.out.println(ret);
        } catch (Exception e) {
            System.err.println(e.toString());
        }
    }
}
```

其中<http://localhost:8091/example>为流程部署的地址，可根据实际情况修改。

客户端调用

首先在环境变量 CLASSPATH 中加入%Engine_HOME%\example\client\lib 下的类包，然后在 windows 下执行下列命令调用 example 流程：

```
C:\> cd xservice-workflow-engine\example\client
C:\ xservice-workflow-engine\example\client> javac ExampleClient.java
C:\ xservice-workflow-engine\example\client> java ExampleClient
abcd
```