

Web Services Example

Web Service Provider Example - Enabling Visible Business

Company A makes earrings. One of their suppliers, Company B, provides the glass beads that are used in the earrings.

Company A has decided that the most efficient way to do business with Company B is to have them check inventory levels through a Web service every night. Company A wants Company B to have real time access to their inventory levels and orders, so that Company B can plan their factory work more efficiently (how many shifts to schedule for the next two weeks, for example). Company A would like Company B to automatically ship them new stock as inventory levels reach certain points. By using Web services, both companies will be able to plan their production and ordering for the upcoming weeks more efficiently.

Company A creates a Web services group called "Inventory_Check." The Web services group includes the following:

Component	Name
Business Process	Bead_Inquiry.bp
Service	Lightweight JDBC adapter configuration named "Bead_LTWTJDBC" (used in the business process)
User Account	One new Gentran Integration Suite user account was created: Bead_User

The following activities happen between Company A and Company B:

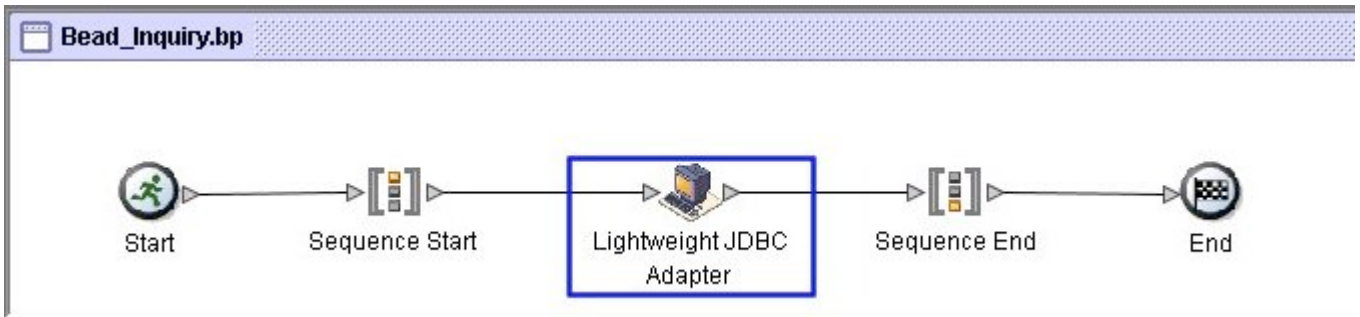
1. Company A creates each of the components listed in the previous table.
2. Company A creates the Web services group, Inventory_Check, then adds the business process and the user account to it.

Note: It is not necessary to add the Lightweight JDBC adapter configuration, Bead_LTWTJDBC, to the Web services group because it is part of the business process that is being added to the Web services group.

3. Company A generates the WSDL for the Web services group and sends it to Company B (or sends them the URL for the WSDL).
4. Company B accesses the WSDL URL and uses the WSDL as information when creating the query that will be sent to the Web service.
5. Company B creates the query that will be automatically sent each night at 2 a.m. As part of the query, they will include values for two fields in the database: part_number and level.
6. When the query is sent to the Web service URL, the Web services invokes the Bead_Inquiry business process. The business process runs the query against the database and returns the response to the Web service. The Web service returns the response to Company B.

Bead_Inquiry Business Process

The following illustration shows how the business process might look in the GPM:



GPM Parameters, Part 1

Service Editor - Lightweight JDBC Adapter

Name: Lightweight JDBC Adapter

Config: Bead_LTWT.JDBC

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: LightweightJDBCAdapterTypeInputMessage

Name	Value	Use XPATH?
InitialWorkflowId	[Not Applicable]	<input type="checkbox"/>
param1	//part_number/text()	<input checked="" type="checkbox"/>
param10		<input type="checkbox"/>
param11		<input type="checkbox"/>
param12		<input type="checkbox"/>
param13		<input type="checkbox"/>
param14		<input type="checkbox"/>
param15		<input type="checkbox"/>
param16		<input type="checkbox"/>
param17		<input type="checkbox"/>
param18		<input type="checkbox"/>
param19		<input type="checkbox"/>
param2	//level/text()	<input checked="" type="checkbox"/>

Takes the two parameters (part_number and level) from process data.

GPM Parameters, Part 2

Name	Value	Use XPATH?
paramtype1	String	<input type="checkbox"/>
paramtype10		<input type="checkbox"/>
paramtype11		<input type="checkbox"/>
paramtype12		<input type="checkbox"/>
paramtype13		<input type="checkbox"/>
paramtype14		<input type="checkbox"/>
paramtype15		<input type="checkbox"/>
paramtype16		<input type="checkbox"/>
paramtype17		<input type="checkbox"/>
paramtype18		<input type="checkbox"/>
paramtype19		<input type="checkbox"/>
paramtype2	Integer	<input type="checkbox"/>
paramtype20		<input type="checkbox"/>

Specifies the parameter types for part_number and level parameters

GPM Parameters, Part 3

Name	Value	Use XPath?
paramtype20		<input type="checkbox"/>
paramtype3		<input type="checkbox"/>
paramtype4		<input type="checkbox"/>
paramtype5		<input type="checkbox"/>
paramtype6		<input type="checkbox"/>
paramtype7		<input type="checkbox"/>
paramtype8		<input type="checkbox"/>
paramtype9		<input type="checkbox"/>
pool	mysqlPool	<input type="checkbox"/>
query_type	Select	<input type="checkbox"/>
result_name	result	<input type="checkbox"/>
row_name	row	<input type="checkbox"/>
sql	SELECT * FROM INV_DB WHERE PART_NUMBER=? AND LEVEL < ?</td> <td><input type="checkbox"/></td>	<input type="checkbox"/>
StartNewWorkFlow	No	<input type="checkbox"/>

Defines the query to be run on the database. The query uses the values passed from process data for part_number and level. It returns records for part_number where inventory is less than the value for level.

The following BPML is the source for the preceding GPM example:

```

<process name = "Bead_inquiry">
  <sequence>
    <operation name="Lightweight JDBC Adapter">
      <participant name="Bead_LTWTJDBC"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1"//part_number/text()</assign>
        <assign to="param2"//level/text()</assign>
        <assign to="paramtype1">String</assign>
        <assign to="paramtype2">Integer</assign>
        <assign to="pool">mysqlPool</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="result_name">result</assign>
        <assign to="row_name">row</assign>
        <assign to="sql">SELECT * FROM INV_DB WHERE PARTNO=? AND LEVEL
&lt;
?&lt;/assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

XML Input File for Query

The following example shows how the XML input file from company B might look:

```

<?xml version="1.0" encoding="UTF-8"?>
<A_Query>
  <part_number>12345</part_number>
  <level>10</level>
</A_Query>

```

WSDL for Inventory_Check Web Services Group

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="Inventory_Check"
  targetNamespace="http://www.sterlingcommerce.com/mesa"

```

```

xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:mesa="http://www.sterlingcommerce.com/mesa"
xmlns:mesa_xsd="http://www.sterlingcommerce.com/mesa/schema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xs:schema attributeFormDefault="unqualified"
      elementFormDefault="qualified"

targetNamespace="http://www.sterlingcommerce.com/mesa/schema"
  xmlns="http://www.sterlingcommerce.com/mesa/schema"
  xmlns:tns="http://www.sterlingcommerce.com/mesa/schema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="Binary">
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="href" type="xs:anyURI"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:element name="attachment" type="tns:Binary"/>
  <xs:element name="inlineAttachment"
type="xs:base64Binary"/>
  <xs:complexType name="ProcessData">
    <xs:sequence>
      <xs:any/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="mesaFault" type="tns:MESAFault"/>
  <xs:complexType name="MESAFault">
    <xs:sequence>
      <xs:element name="code"/>
      <xs:element name="message"/>
      <xs:element name="statusReport"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="processData" type="tns:ProcessData"/>
  <xs:element name="documents">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded"
ref="tns:attachment"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="HashType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="MD5"/>
      <xs:enumeration value="NONE"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="MESAAuth">
    <xs:sequence>
      <xs:element name="principal"/>
      <xs:element name="auth">

```

```

        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="hashType"
type="tns:HashType"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:element name="mesaAuth" type="tns:MESAAuth"/>
<xs:element name="Bead_inquiry" type="tns:ProcessData"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="MESAResponse">
    <wsdl:part element="mesa_xsd:processData" name="parameters"/>
    <wsdl:part element="mesa_xsd:attachment" name="attachment"/>
</wsdl:message>
<wsdl:message name="Bead_inquiry">
    <wsdl:part element="mesa_xsd:mesaAuth" name="header"/>
    <wsdl:part element="mesa_xsd:Bead_inquiry" name="parameters"/>
    <wsdl:part element="mesa_xsd:attachment" name="attachment"/>
</wsdl:message>
<wsdl:message name="MESAFault">
    <wsdl:part element="mesa_xsd:mesaFault" name="parameters"/>
</wsdl:message>
<wsdl:message name="GISGeneric">
    <wsdl:part element="mesa_xsd:mesaAuth" name="header"/>
    <wsdl:part element="mesa_xsd:processData" name="parameters"/>
    <wsdl:part element="mesa_xsd:attachment" name="attachment"/>
</wsdl:message>
<wsdl:portType name="GISPortType">
    <wsdl:operation name="executeBead_inquiry">
        <wsdl:input message="mesa:Bead_inquiry"/>
        <wsdl:output message="mesa:MESAResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="GISBinding" type="mesa:GISPortType">
    <soap:binding
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="executeBead_inquiry">
        <soap:operation soapAction="sii:Bead_inquiry"/>
        <wsdl:input>
            <mime:multipartRelated>
                <mime:part>
                    <soap:body parts="header parameters"
use="literal"/>
                </mime:part>
                <mime:part>
                    <mime:content part="attachment"
type="application/octetstream"/>
                </mime:part>
            </mime:multipartRelated>
        </wsdl:input>
        <wsdl:output>

```

```
        <mime:multipartRelated>
            <mime:part>
                <soap:body parts="parameters" use="literal"/>
            </mime:part>
            <mime:part>
                <mime:content part="attachment"
type="application/octetstream"/>
            </mime:part>
        </mime:multipartRelated>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Inventory_Check">
    <wsdl:port binding="mesa:GISBinding" name="GISPort">
        <soap:address location="null?service=Inventory_Check"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Creating or Editing a Web Services Group

Name and Description

Parameter	Description
Name	Unique name for the Web services group. Required.
Description	Description for the Web services group. Required.

Security Settings

Parameter	Description
Request Settings: Verification Certificate	Enter the name of a verification certificate (CA or trusted certificate). To access the list of available certificates, click the List icon to the right of the field. Select a certificate from the list and click Save . Optional.
Response Settings: Signing Certificate	Enter the name of a signing certificate (system certificate). To access the list of available certificates, click the List icon to the right of the field. Select a certificate from the list and click Save . Optional.

Assign Business Processes

Parameter	Description
Filter by name	To filter the list of available business processes, type part of the business process name into the Filter field and click the Filter icon. Optional.
Available and Selected lists	Select one or more business processes from the list of available business processes on the left to be associated with this Web services group and click the right arrow. To select all available business processes, click the double right arrow. Optional.

Assign Services

Parameter	Description
Filter by name	To filter the list of available services, type part of the service name into the Filter field and click the Filter icon.
Available and Selected lists	Select one or more services from the list of available services on the left to be associated with this Web services group and click the right arrow. To select all available services, click the double right arrow.

Assign Consumers

Parameter	Description
Filter by name	To filter the list of available users, type part of the user name into the Filter field and click the Filter icon.
Available and Selected lists	Select one or more users from the list of available users on the left to be associated with this Web services group and click the right arrow. To select all available services, click the double right arrow.

Confirm

Confirm your selections and click **Finish**.

Once the group has been created, the following message displays:

The base URL for your Web service is **http://servername:00000**

Where *servername* is the name of the server and *00000* is the port number that you should reference for this Web services group.

Next, generate the WSDL for this Web services group.

The base URL for your web service is <http://10.2.32.24:9040>

The base URL for your web service is <http://10.2.32.24:9040>